## Applied Accelerated Artificial Intelligence Dr. Satyajit Das Department of Computer Science and Engineering Indian Institute of Technology, Palakkad

## Lecture - 29 Accelerated TensorFlow - XLA Approach Part - 2

(Refer Slide Time: 00:15)



So, now first we will talk about how to use TensorFlow for your TPU environment ok. So, this is the very very important for your TPU cluster if you want to use TPU clusters. So, basically for your colab you can go to your runtime and change runtime type to your as simple as that. So, basically in by default TPUs will be available for you and you can use that.

Now, how we can use that and we will use that with different strategies ok. So, basically XLA mixed precision tf.function run time and so on distribution strategy right. So, let us see all of them in one place ok. So, importing the things here, so all the things are imported and then you just see which profiler which TPU version is being used we are just printing here and using auto time for time displays just few libraries.

(Refer Slide Time: 01:31)



Now, checking if TPU is available and changing the runtime to your TPU. So, basically distribute.

Student: Sir.

TPU cluster resolver, yes.

Student: Sir we cannot view the text properly. So, requesting to increase the size.

Yes. So, I hope this is visible now. So, till now whatever we have talked about is basically the imports and setup that is very simple, but now what we are going to see is that how to define the how to use the TPU for your devices and distribution strategies and so on.

So, we are defining the resolver how we are defining here TPU cluster resolver that we have seen in the previous class experimental connect to cluster we are connecting it and then TPU initialization code right. So, to just at the beginning you need to initialize your TPUs. Now by the way XLA is very good for your TPUs ok because many fused operations that you are you the XLA will generate is directly supported onto TPU because TPU is basically from Google TensorFlow is from Google XLA is from Google. Now, you connect everything ok.

(Refer Slide Time: 03:17)

00	🐜 🖕 VIDHYA XLA TPU.ipynb 🕆	
c	File Edit View Insert Runtime Tools Help Last saved at 01:11	Comment 🗳 Share 🌣 💟
=	+ Code + Text	Connect 👻 🖍 Editing 🔺
-	INFU:tensortlow:Finished initializing IPU system.	
٩	<ul> <li>JNF0:tensorflow:Finished initializing TPU system. Found TPU at: [LogicalDevice[mame='/job:worker/replica:0/task:0/device:TI time: 12.2 s (started: 2022-03-02 14:58:15 +00:00)</li> </ul>	PU:0', device_type='TPU'), LogicalDevice(name='/job:worker/r
0	4	
		↑↓◎Ц┇Ы:
{x}	def my func vla(a b c):	
	return tf. reduce sum(a + b * c)	
n		
-	time: 2.36 ms (started: 2022-03-02 14:58:30 +00:00)	
	<pre>with tf.device('/TPU:0'):</pre>	
	<pre>with tf.compat.vl.Session() as sess_w_xla: print(sess_w_xla.run(my_func_xla(tf.ones([4,4]),tf.ones([4,</pre>	es([4,4]))))
	r. 32.0	
	time: 424 ms (started: 2022-03-02 14:58:34 +00:00)	
	[ ] strategy = tf.distribute.TPUStrategy(resolver)	
	INFO:tensorflaw:Found TPU system:	
	INFO:tensorflow:Found TPU system:	
	INFO:tensorflow:*** Num TPU Cores: 8	
E	INFO:tensorflow:*** Num TPU Cores: 8	
-		

We will see what kind of performance we are getting here tf.function JIT compiler equal to True.

(Refer Slide Time: 03:45)



So, this is how we will enable here now first we are just wrapping up with some function. So, this is one function that we are going to reduce. So, basically this is the graph computational graph and we are running it with the TPU that is it right.

So, all the TPU information you are getting and then you can see the runtime here and the strategy that we have to defined before right. So, the strategy we have defined here distribute strategy for your TPU and we are using that strategy for running this particular function that we have created.

(Refer Slide Time: 04:06)

File Edit View Insert Runtime Tools Help Last saved at 01:11 + Code + Text	Connect · Editing
[] @tf.function(jit_compile=True)	
return tf.reduce sum(a + b * c)	
ALTER 2 2 (-ALTER 2022 02 20 10 10 10 10 10 10 10	
time: 2.2 ms (started: 2022-03-02 14:35:41 +00:00)	
A z = strategy run(my func via args=/if ones/(4 41) tf ones/(4 41) tf ones/(4 41)))	<u>↑↓©Щ₽Ы∎:</u>
print(z)	
D. PerBenlicar/	
0: tf.Tensor(32.0, shape=(), dtype=float32),	
1: tf.Tensor(32.0, shape=(), dtype=float32), 2: tf Tensor(32.0, shape=(), dtype=float32)	
3: tf.Tensor(32.0, shape=(), dtype=float32),	
4: tf.Tensor(32.0, shape=(), dtype=float32), 5: tf.Tensor(32.0, shape=(), dtype=float32).	
6: tf.Tensor(32.0, shape=(), dtype=float32),	
<pre>/: tf.lensor(32.0, shape=(), dtype=float32) }</pre>	
time: 1.11 s (started: 2022-03-02 14:58:43 +00:00)	
MNIST MODEL WITHOUT TPU AND WITHOUT XLA	

So, this is a very simple example how you will actually create the strategy enable the XLA compiler and run ok.

(Refer Slide Time: 04:19)



But let us take a concrete example for model training where you will apply all this. So, basically here what you are seeing that all the necessary imports and version checking

and that is fine. Again, TensorFlow input for the data set because we will use tfds for importing the data sets.

(Refer Slide Time: 04:34)



Whatever the data set that we want to load here we are loading mnist datasets splitting train test and shuffle True supervised True all these things we are setting up good.

(Refer Slide Time: 04:54)



Then we are normalizing the data and setting up one input pipeline. So, that we talked about previous section also that map is basically transformation with this normalized function right. And the number of parallel calls we are letting how many number of parallel calls that will be generated.

(Refer Slide Time: 05:32)



ds\_train we are getting the cache enable, shuffle batching prefetching everything is there. So, you now know how to do that now build an evaluation pipeline. So, basically just checking here and creating the model here for the sequential model simply we are creating three layers of model one flattening layer that is just taking the input and flattening it two dense layers are there and this is the output, output of that is soft max function ok. (Refer Slide Time: 05:58)

-	C VIDHTA_ALA_IPU.Ipynb 12		🗖 Comment 😃 Share 🏚 🕥
	File Edit View Insert Runtime Tools Help Last saved at 01	11	
+	Code + Text		Connect 👻 🎤 Editing 🔷
1	flatten_7 (Flatten) (None, 784)	0	↑↓⇔■‡↓]≣ :
	C+ dense 14 (Dense) (None 128)	166488	
	dense_14 (bense) (none, 120)	100400	
	dense_15 #(Dense) (None, 10)	1290	
	Total params: 101,770		
	Trainable params: 101,770		
	Non-crainable parails. 0		
	time: 79.6 ms (started: 2022-03-02 15:51:51 +00	:00)	
1	<pre>] # Training without XLA Compiler # To train a model with fit(), you need to spec # For regression models, the commonly used loss # while for classification models predicting th</pre>	ify a loss function, an optimizer, function used is mean squared err e probability, the loss function m so function for multi-class class;	, and optionally, some metrics to monitor. for function nost commonly used is cross entropy. fication model where the output label is assigned
	<pre># sparse_categorical_crossentropy: Used as a lo # A metric is a function that is used to judge # except that the results from evaluating a met tf.config.optimizer.set_jit(False) # Start with</pre>	the performance of your model. Met ric are not used when training the XLA disabled.	tric functions are similar to loss functions, e model.
	# sparse_categorical_crossentropy: Used as a LC # A metric is a function that is used to judge # except that the results from evaluating a met tf.config.optimizer.set jit(False) # Start with # @tf.function - RuntimeError: Detected a call	the performance of your model. Met ric are not used when training the XLA disabled. to 'Model.fit' inside a 'tf.functi	rric functions are similar to loss functions, e model. ion'. 'Model.fit is a high-level endpoin
	<pre># sparse_categorical_crossentropy: Used as a la # A metric is a function that is used to judge # except that the results frame wealuating a met tf.config.optimizer.set_jit(False) # Start with # @tf.function - RuntimeError: Detected a call # Please move the call to "Model.fit' outside to # Please move the call to "Model.fit' outside to</pre>	the performance of your model. Met ric are not used when training the XLA disabled. to 'Model.fit' inside a 'tf.functi f all enclosing 'tf.function's.	tric functions are similar to loss functions, e model. Lon". 'Model.fit is a high-level endpoin

So, model.summary () as you can see here we have three layers here and the summary.

(Refer Slide Time: 06:07)



Now, at first you just to see how it is performing we are setting the tf.config.optimizer.set \_ jit (False ), this is another way to set JIT compiler equal to True or set JIT compiler equal to False. So, we are disabling XLA because we have enabled previously for that toy example learning then model equal to compile. So, basically, we are compiling the model.

(Refer Slide Time: 06:34)



And now we are creating the training pipeline for that and we are using we will be using since you can see that here we have generated the model using the sequential api. So, it is I mean highly likely that you will use model.fit () ok.

So, but as you can see here as tf.function (jit\_compile = True) is essentially trying to make the computational graph and model.fit() is actually again you are doing that, but; that means, tf.function runtime is not supported for your model of fit.

(Refer Slide Time: 07:32)



So, if you are using model of fit you will not be able to enable the XLA. So, just importing the tf and data set and running the resolver initiating the devices right.

(Refer Slide Time: 07:41)



(Refer Slide Time: 07:47)



So, all these for TPU that you have seen so far. Just getting some data about the image size and image classes I can see 60000 images up there for your train and test data is 10000 and so forth.

(Refer Slide Time: 07:55)



And then this data input pipeline normalize which is used in map in cache shuffle repeat batch prefetch.

(Refer Slide Time: 08:06)



And then we are building the evaluation pipeline now. So, basically calling the functions here again using the tf.device ().

(Refer Slide Time: 08:16)

File Edit Vere Innett Runtime Tools Help LastEmediat10:111         # + Code + Text       Connect • / / / / / / / / / / / / / / / / / /		Comme	ent 🚢	Share	۵	V
<pre>= + Code + Text Connect • ✓     # Context = 28*28     # Engle Times Size = 28*28     # Engle Times Size = 28*28     # Engle Times Size = 28*28     # Dense Layer:     # Engle = 784     # Dense Layer:     # Training with XLA Compiler     # Training with XLA Compiler     # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor     # For regression models, the commonly used loss function used is man squared error function     # for regression models, the commonly used loss function for multi-class classification model where the output Label is     # sparse, categorical, crossentropy: Used as a loss function for multi-class classification model where the output Label is     # A metric is a function that is used to Ugade the performance of your model. Metric functions are similar to loss funct     # except that the results from evaluating a metric are not used when training the model.     tf.config.optimizer.set jit(True) # Emple XLA.     # fit.function - RuntimeError: Detected a call to 'Model.fit' inside a 'tf.function'. 'Kodel.fit is a high-level endpoint</pre>						-
<pre></pre>		Cor	nnect 👻	18	Editing	^
<pre>9 # Flatten Layer: 28 * 28 = 784 # # Entre Layer: 28 * 28 = 784 # Dente Layer: # Input = 784 # Output = 128 (As mentioned in code) # Paras # = 784 * 128 + 128 (bias) = 100480 # Similarly Next Layer # Training with XLA Compiler # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # for regression models, the commonly used loss function sed is mean squared error function # for regression models, the commonly used loss function for multi-class classification model where the output label is # A metric is a function that is used to judge the performance of your model. Metric functions are similar to loss funct # config.optimizer.set jit(True) # Emple XLA. # dit.function - RuntimeError: Detected a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the compiler is a common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # the common output a call to 'Model.fit' inside a 'tf.function'. 'Model.fit' inside a 'tf.function'. 'Model.fit' inside a 'tf.function'. 'Model.fit' inside</pre>			h		0.0	1.1
<pre>4 @ Dense Layer: # I papt = 784 4 Output = 128 (As mentioned in code) # Paras # = 784 * 128 + 128 (As mentioned in code) # Paras # = 784 * 128 + 128 (As mentioned in code) # Similarly Next Layer # Training with XLA Compiler # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # For regression models, the commonly used loss function used is man squared error function # while for classification models predicting the probability, the loss function most commonly used is cross entropy. # sparse categorical crossentropy: Used as a loss function for multi-class classification model where the output label is # A metric is a function that is used to judge the performance of your model. Netric functions are similar to loss functi # except that the results from evaluating a metric are not used when training the model. tf.config.optimizer.set jit(True) # Emple XLA. # tit.function - RuntimeError: Detected a call to 'Model.fit' inside a 'tf.function'. 'Rodel.fit is a high-level endpoint</pre>		-	4 65		12 I	
<pre>&gt; # Input = 784 # Duput = 128 (As mentioned in code) # Paras # = 784 * 128 + 128 (bias) = 109480 # Similarly Next Layer # Training with XLA Compiler # for train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # for regression models, the commonly used loss function used is mean squared error function # while for classification models predicting the probability, the loss function most commonly used is cross entropy. # sparse categorical crossentropy. Used as a loss function for multi-class classification model where the output label is # A metric is a function that is used to judge the performance of your model. Metric functions are similar to loss funct # except that the results frame evaluating a metric are not used when training the model. tf.config.optimizer.set jit(True) # Emple XLA. # dtf.function - RuntimeError: Detected a call to 'Model.fit' inside a 'tf.function'. 'Rodel.fit is a high-level endpoint</pre>						
<pre>c&gt; # Output = 128 (As mentioned in code) # Param # 784 + 128(bias) = 109480 # Similarly Next Layer # Training with XLA Compiler # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # To train a model with fit(), you need to specify a loss function for white for classification models greated error function # white for classification models ymetric ing the probability, the loss function most somenly used is cross entropy. # sparse categorical, crossentropy: Used as a loss function for multi-class classification model where the output label is # A metric is a function that is used to judge the performance of your model. Netric functions are similar to loss funct # except that the results from evaluating a metric are not used when training the model. tf.config.optimizer.set jit(True) # Emple XLA. # tit.function - RuntimeError: Detected a call to 'Nodel.fit' inside a 'tf.function'. 'Nodel.fit is a high-level endpoint # the training the model. # the training the space are not used a call to 'Nodel.fit' inside a 'tf.function'. 'Nodel.fit is a high-level endpoint # the training the</pre>						
<pre># Para # = PA* + 128 + 128 (bias) = 100460 # Similarly Next Layer # Training with XLA Compiler # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # To train a model with fit(), you need to specify a loss function for compily used is cross entropy. # sparse categorial, crossentropy. Used as a loss function for multi-class classification model where the output label is # notic is a function that is used to judge the performance of your model. Netric functions are similar to loss funct # except that the results frme evaluating a metric are not used when training the model.     tf.config.optimizer.set jit(True) # Emple XLA. # dit.function - RuntimeError: Detected a call to 'Nodel.fit' inside a 'tf.function'. 'Rodel.fit is a high-level endpoint # displacement of the set of t</pre>						
<pre>[1] # Similarly Next Layer # Training with XLA Compiler # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # For regression models, the commonly used loss function used is man squared error function # while for classification models predicting the probability, the loss function most commonly used is cross entropy. # sparse categorical, crossentropy: Used as a loss function for multi-class classification model where the output label is # A metric is a function that is used to Ugde the performance of your model. Netric functions are similar to loss functi # except that the results from evaluating a metric are not used when training the model. tf.config.optimizer.set_jit(True) # Emple XLA. # dtf.function - RuntimeError: Detected a call to 'Nodel.fit' inside a 'tf.function'. 'Nodel.fit is a high-level endpoint</pre>						
# Training with XLA Compiler # Training with XLA Compiler # To train a model with fit(), you need to specify a less function, an optimizer, and optionally, some metrics to monitor # For regression models, the commonly used loss function used is mean squared error function # while for classification models predicting the probability, the loss function most commonly used is cross entropy. # sparse categorical crossentropy: Used as a loss function for multi-class classification model where the output label is # A metric is a function that is used to judge the performance of your model. Netric functions are similar to loss funct # except that the results from evaluating a metric are not used when training the model. tf.config.optimizer.set jit(True) # Emple XLA. # dtf.function - RuntimeError: Detected a call to 'Nodel.fit' inside a 'tf.function'. 'Rodel.fit is a high-level endpoint						
# Training with XLA Compiler # Training with XLA Compiler # To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # For representation models, the commonly used loss function used is mean squared error function # while for classification models predicting the probability, the loss function nost commonly used is cross entropy. # sparse categorical, crossentropy: Used as a loss function for multi-class classification model where the output label is # A metric is a function that is used to judge the performance of your model. Netric functions are similar to loss function # except that the results from evaluating a metric are not used when training the model. tf.config.optimizer.set_jit(True) # Emple XLA. # tf.function - RuntimeError: Detected a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint						
<pre># To train a model with fit(), you need to specify a loss function, an optimizer, and optionally, some metrics to monitor # For regression models, the commonly used loss function used is man squared error function # while for classification models predicting the probability, the loss function most emorphy used is cross entropy. # sparse categorical crossentropy: Used as a loss function for multi-class classification models where the output label is # A metric is a function that is used to judge the performance of your model. Netric functions are similar to loss funct # except that the results from evaluating a metric are not used when training the model. tf.config.optimizer.set jit(True) # Emple XLA. # dtf.function - RuntimeError: Detected a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint</pre>						
<pre># Io Train a model with Ti(1), you need to specify a loss function, an optimizer, and optimally, some extrice to monitor # For regression models, the commonly used loss function used is men squared error function # while for classification models predicting the probability, the loss function model with the output label is # A metric is a function that is used to judge the performance of your model. Metric functions are similar to loss functi # accept that the results from evaluating a metric are not used when training the model.     tf.config.optimizer.set_jit(True) # Emple XLA. # @tf.function - RuntimeError: Detected a call to 'Model.fit' inside a `tf.function'. 'Model.fit is a high-level endpoint </pre>						
<pre># For regression models, the commonly used loss function used is seam squared error runction # while for classification models predicting the probability, the loss function most commonly used is cross entropy. # sparse categorical crossentropy: Used as a loss function for multi-class classification model where the output label is # A metric is a function that is used to judge the performance of your model. Netric functions are similar to loss functi # except that the results from evaluating a metric are not used when training the model. tf.config.optimizer.set jit[True] # Emple XLA. # tf.function - RuntimeError: Detected a call to 'Bodel.fit' inside a 'tf.function'. 'Rodel.fit is a high-level endpoint # continue in the instruction of the second secon</pre>	lionally, s	, some metri	ics to m	aonitor	•	
<pre># while for classification models predicting the preaduality, the loss function most commonly used is cross entropy. # sparse categorical crossentropy: Used as a loss function for multi-classification model, where the output label is # A metric is a function that is used to judge the performance of your model. Metric functions are similar to loss functi # except that the results from evaluating a metric are not used when training the model. tf.config.optimizer.set_jit[(True) # Emple XLA. # @ff.function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'. 'Model.fit is a high-level endpoint </pre>	110n					
<pre># sparse_categorical_crossentropy: used as a uses function for multi-class classification mode. Where the output cake. If # A metric is a function that is used to Updge the performance of your model. Netric functions are similar to loss functi # except that the results from evaluating a metric are not used when training the model. tf.config.optimizer.set jit[True] # Emple XLA. # @tf.function - RuntimeError: Detected a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint # output call to the set of the</pre>	ionly used	ed 1s cross	entropy	1.		
<pre># A metric is a function that is used to judge the periofmance of your model. metric inclues are similar to loss funct # except that the results frem evaluating a metric are not used when training the model. tf.config.optimizer.set jit(True) # Emple XLA. # gtf.function - RuntimeError: Detected a call to 'Rodel,fit' inside a 'tf.function'. 'Rodel.fit is a high-level endpoint</pre>	I nodel whe	where the ou	stput la	abet 15	assig	gned
<pre># except that the results from evaluating a metric are not used when training the model. tf.config.optimizer.set_jit(True) # Emple XLA. # @ff.function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'. 'Model.fit is a high-level endpoint # off.function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'. 'Model.fit is a high-level endpoint # off.function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'. 'Model.fit is a high-level endpoint # of the function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'. 'Model.fit is a high-level endpoint # off.function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'. 'Model.fit is a high-level endpoint # off.function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'. 'Model.fit is a high-level endpoint # off.function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'. 'Model.fit is a high-level endpoint # off.function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'. 'Model.fit is a high-level endpoint # off.function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'. 'Model.fit is a high-level endpoint # off.function - RuntimeError: Detected a call to 'Model.fit' inside a 'ff.function'.'# off.function'.'# of</pre>	tions are	re similar t	to Loss	TUNCTI	ons,	
<pre>tf.config.optimizer.set jit(True) # Emple XLA. # gtf.function - RuntimeError: Detected a call to 'Rodel.fit' inside a 'tf.function'. 'Rodel.fit is a high-level endpoint</pre>						
<pre># dtf.function - RuntimEFror: Detected a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint</pre>						
# @tf.function - RuntimeError: Detected a call to 'Model.fit' inside a 'tf.function'. 'Model.fit is a high-level endpoint						
The second secon	del fit is	is a high-1	level er	ndnoint	that	mana
# Please nove the call to model tit outside of all enclosing it function s	Nettine as	as a magnet	teret of	rapozite	ende	ing the
# Note that you can call a 'Model' directly on 'Tensor's inside a 'tf.function' like: 'model(x)'.						
# We have to customer the training loon	(v) [ohu					
	ndel(x)'.					-
model.compile(	odel(x)'.					
losse categorical concentrany'	odel(x)`.					
LUSS = SUDISC LOLCUUIILOL LIUSSCILLUUV .	odel(x) <sup>°</sup> .			- 1		

But here now we are enabling the JIT set JIT equal to True. So, if you see the earlier training for that matter.

(Refer Slide Time: 08:29)



So, if you see here where we are actually using the modeling ok. So, where we have repeated yeah ok. So, ok this is the model summary and yeah.

#### (Refer Slide Time: 08:53)

co	La VIDHYA_XLA_TPU.ipynb ☆	🖪 Comment 🛛 🛤 Share 🏚 🚺	)
	File bolt view insert kuntime loois Help Lastsaved at 01:11	2	
= _ ·	r code i r reat	connect • / counting	
	<pre>with tf.device('/TPU:0'):</pre>	^↓☺■‡₽:	
٩	<pre>model = tf.keras.models.Sequential([</pre>		
	tf.keras.layers.Flatten(input_shape=(28, 28, 1)), # A Flatten layer in Keras re	eshapes the tensor to have a shape that is equal to	
0	<pre>tf.keras.layers.Dense(128,activation='relu'), # Dense implements the operation:</pre>	: output = activation(dot(input, kernel) + bias) -	
	<pre>tf.keras.layers.Dense(NUM_CLASSES, activation='softmax')</pre>		
{x}	1)		
	model summary()		
D	# Input Image Size = 28*28		
	# Flatten Layer = 28 * 28 = 784		
	# Dense Layer:		
	# Input = 784		
	<pre># Output = 128 (As mentioned in code)</pre>		
	# Param # = /84 * 128 + 128(blas) = 100480		
	* Similarly Next Layer		
	# Training with XLA Compiler		
	# To train a model with fit(), you need to specify a loss function, an optimizer,	, and optionally, some metrics to monitor.	
	# For regression models, the commonly used loss function used is mean squared err	ror function	
	# while for classification models predicting the probability, the loss function m	most commonly used is cross entropy.	
	<pre># sparse_categorical_crossentropy: Used as a loss function for multi-class classi</pre>	ification model where the output label is assigned	
	# A metric is a function that is used to judge the performance of your model. Met	tric functions are similar to loss functions	
	# except that the results from evaluating a metric are not used when training the	e model.	

### (Refer Slide Time: 08:56)



(Refer Slide Time: 09:02)



So, set JIT True and compile. And then we are using the distributed strategy ok TPU strategy, because we want to distribute all the TPUs that is available here. And thus inside this strategy scope we are defining the model sequential.

(Refer Slide Time: 09:16)



(Refer Slide Time: 09:24)

co	EVIDHYA_XLA_TPU.ipynb     ☆     File Edit View Insert Runtime Tools Help Lastsaved at 01:11	Comment	-	Share	\$	V
	+ Code + Text	Conne	<b>t</b> •	/ Er	liting	^
0	# except that the results from evaluating a metric are not used when training the model.	<u></u>	0	9 \$	0 1	1
<> {x}	# @ff.function - RuntimEFror: Detected a call to 'Model.fit' inside a 'tf.function'. 'Model. # Please move the call to 'Model.fit' outside of all enclosing 'tf.function's. # Note that you can call a 'Model' directly on 'Tensor's inside a 'tf.function' Like: 'model! # Ne have to customize the training loop	.fit is a high-lev (x)`.	el end	dpoint	that m	tana
0	<pre>model.comple( loss='sparse_cateporical_crossentropy', optimizer=ff.keras.optimizers.Adam(), metrics=['accuracy'] ]</pre>					T
	D. NF0:tessorflow:Found TPU system: INF0:tessorflow:Found TPU system: INF0:tessorflow:Fwin TPU Cores: 8 INF0:tessorflow:FWin TPU Cores: 8 INF0:tessorflow:FWI Nu TPU Cores: FWI Cores: FWI FUNCTION INF0:tessorflow:FWI Nu TPU Cores FWI Cores: 8 INF0:tessorflow:FWI Nu TPU Cores FWI Core: 8 INF0:tessorflow:FWI Nu TPU Core: 8 INF0:tessorflow	ce:CPU:0, CPU, 0, ce:CPU:0, CPU, 0, CPU:0, CPU, 0, 0) CPU:0, CPU, 0, 0) DHA TON: A AN	8) 9)			6

And after that we are generating the model summary and setting up the compiler here ok XLA True.

(Refer Slide Time: 09:26)

	File		🗖 Comment 🔐 Share 🏚 😡
		Edit View Insert Runtime Tools Help Last saved at 01:11	• •
	+ Cod	le + Text	Connect 👻 🧨 Editing 🗠
	0	loss='sparse categorical crossentropy',	↑↓∞ <b>□</b> ¢∬∎:
	~	optimizer=tf.keras.optimizers.Adam(),	
		metrics=['accuracy']	
>		)	1
3	D+	INFO:tensorflow:Found TPU system:	
		INFO:tensorflow:Found TPU system:	
		INFO:tensorflow:*** Num TPU Cores: 8	
		INFO:tensorflow:*** Num TPU Cores: 8	
		INFO:tensorflow:*** Num TPU Workers: 1	
		INFO:tensorflow:*** Num TDU Cores Dar Warker: 8	
		INFO:tensorflow:*** Num TPU Cores Per Worker: 8	
		INFO:tensorflow:*** Available Device: DeviceAttributes(/job:localhost/replica:0/	/task:0/device:CPU:0, CPU, 0, 0)
		<pre>INF0:tensorflow:*** Available Device: DeviceAttributes(/job:localhost/replica:0/ INF0:tensorflow:*** Available Device: DeviceAttributes(/job:localhost/replica:0/</pre>	/task:0/device:CPU:0, CPU, 0, 0) /task:0/device:CPU:0, CPU, 0, 0)
		INF0:tensorflow:*** Available Device: DeviceAttributes//job:localhost/replica:0/ INF0:tensorflow:*** Available Device: DeviceAttributes//job:localhost/replica:0/ INF0:tensorflow:*** Available Device: DeviceAttributes//job:worker/replica:0/tas	/task:0/device:CPU:0, CPU, 0, 0) /task:0/device:CPU:0, CPU, 0, 0) sk:0/device:CPU:0, CPU, 0, 0)
		INFO:tensorflow:*** Available Device: DeviceAttributes//job:localhost/replica:0/ INFO:tensorflow:*** Available Device: DeviceAttributes//job:localhost/replica:0/ INFO:tensorflow:*** Available Device: DeviceAttributes//job:worker/replica:0/tas INFO:tensorflow:*** Available Device: DeviceAttributes//job:worker/replica:0/tas	/task:0/device:CPU:0, CPU, 0, 0) /task:0/device:CPU:0, CPU, 0, 0) sk:0/device:CPU:0, CPU, 0, 0) sk:0/device:CPU:0, CPU, 0, 0)
		IMPGressorTox:** Available Device: DeviceAttributes//jobicalbox/replica/b IDPGressorTox:** Available Device: DeviceAttributes//jobicalbox/replica/b IDPGressorTox:** Available Device: DeviceAttributes//jobicarbox/rep/ica/bas IMPGressorTox:** Available Device: DeviceAttributes//jobicarbox/refreplica/bas IMPGressorTox:** Available Device: DeviceAttributes//jobicarbox/refreplica/bas	/task:0/device:CPU:0, CPU, 0, 0) /task:0/device:CPU:0, CPU, 0, 0) sk:0/device:CPU:0, CPU, 0, 0) sk:0/device:CPU:0, CPU, 0, 0) sk:0/device:TPU:0, TPU, 0, 0)
		INFO:tensorflow:"* Available Device: DeviceAttributes(/jobicalbost/replica%) INFO:tensorflow:"* Available Device: DeviceAttributes(/jobicalbost/replica%) INFO:tensorflow:"* Available Device: DeviceAttributes(/jobarker/replica%) INFO:tensorflow: ** Available Device: DeviceAttributes(/jobarker/replica%)	/task:0/device:(PU:0, (PU, 0, 0) /task:0/device:(PU:0, (PU, 0, 0) sk:0/device:(PU:0, (PU, 0, 0) sk:0/device:(PU:0, (PU, 0, 0) sk:0/device:(PU:0, TPU, 0, 0) sk:0/device:(PU:0, TPU, 0, 0) sk:0/device:(PU:0, TPU, 0, 0)
		IMPG-tensorTox:** Available Device: Devicettributes(/jobicalbst/replica%) IMPG-tensorTox:** Available Device: Devicettributes(/jobicalbst/replica%) IMPG-tensorTox:** Available Device: Devicettributes(/jobicarbst/replica%) IMPG-tensorTox:** Available Device: Devicettributes(/jobicarbstrophica%) IMPG-tensorTox:** Available Device: Devicettributes(/jobicarbstrophica%)	/task:0/device:(PU:0, CPU, 0, 0) /task:0/device:(PU:0, CPU, 0, 0) sk:0/device:(PU:0, CPU, 0, 0) sk:0/device:(PU:0, CPU, 0, 0) sk:0/device:(PU:0, CPU, 0, 0) sk:0/device:(PU:0, TPU, 0, 0) sk:0/device:(PU:0, TPU, 0, 0) sk:0/device:(PU:0, 17U, 0, 0) sk:0/device:(PU:0, 17U, 0, 0)
		INFO:tensorflox:"* Available Device: DeviceAttributes//jobiocalbost/replica?/ INFO:tensorflox:"* Available Device: DeviceAttributes//jobiocalbost/replica?/ INFO:tensorflox:"* Available Device: DeviceAttributes//jobion/ker/replica?/ INFO:tensorflox:"* Available Device: DeviceAttributes//jobion/ker/replica?/ INFO:tensorflox:** Available Device: DeviceAttributes// Jobion/ker/replica?/ INFO:tensorflox:** Available Device:************************************	/task://device:CPU:6, CPU, 0, 0) /task://device:CPU:6, CPU, 0, 0) sk://device:CPU:6, CPU, 0, 0) sk://device:CPU:6, CPU, 0, 0) sk://device:CPU:6, CPU, 0, 0) sk://device:CPU:6, CPU, 0, 0) sk://device:CPU:0, CPU, 0, 0) sk://device:CPU:1, CPU, 0, 0) sk://device:CPU:1, CPU, 0, 0)
		DBG-tensorflor** Available Device. Periodettibutes/jbiolablest/replica/b DBG-tensorflor** Available Device. DBG-tensorflor** Available Device. Periodettibutes/jbiowree/replica/base DBG-tensorflor** Available Device. Periodettibutes/jbiowree/replica/base	//taski/dwice/PRi-8, CPU, 8, 8) /taski/dwice/PRi-8, CPU, 8, 8) sks/dwice/PRi-8, CPU, 8, 0) sks/dwice/PRi-8, CPU, 8, 0) sks/dwice/PRI-8, CPU, 8, 0) sks/dwice/PRI-8, PU, 8, 0) sks/dwice/PRI-1, PU, 8, 0) sks/dwice/PRI-2, PU, 8, 0) sks/dwice/PRI-2, PU, 8, 0) sks/dwice/PRI-2, PU, 8, 0)
		INFO:tensorflox:"* Available Device: Devicettributes//jobiolablost/replica?/ INFO:tensorflox:"* Available Device: Devicettributes//jobioarbest/replica?/ INFO:tensorflox:"* Available Device: Devicettributes//jobioarbestreplica?/ INFO:tensorflox:"* Available Device: Devicettributes//jobioarbestreplica?/ INFO:tensorflox:** Available Device: Devicettributes// INFO:tensorflox:** Available Device: Devicettributes// INFO:tensorflox:** Available Device: Devicettributes//	/taski/dwirec.FDI-8, CFU, 0, 0) /taski/dwirec.FDI-8, CFU, 0, 0) ski/dwirec.FDI-8, CFU, 0, 0) ski/dwirec.FDI-8, CFU, 0, 0) ski/dwirec.FDI-0, FU, 0, 0) ski/dwirec.FDI-0, FU, 0, 0) ski/dwirec.FDI-1, FU, 0, 0) ski/dwirec.FDI-1, FU, 0, 0) ski/dwirec.FDI-1, FU, 0, 0) ski/dwirec.FDI-1, FU, 0, 0)
		DBG-tensorflax"* Anallable Device: DeviceAttributes(/jbiolablest/replica/b) DBG-tensorflax"* Anallable Device: DeviceAttributes(/jbiolablest/replica/b) DBG-tensorflax"* Anallable Device: DeviceAttributes(/jbiowrker/replica/b) DBG-tensorflax"* Anallable Device: DeviceAttributes(/jbiowrker/replica/b)	$\label{eq:constraint} \begin{array}{l} (T_{20}, k_{1}, 0) \in 0, 0 \\ (T_{20}, k_{1}, 0) \in (r_{10}, 0, 0), 0, 0) \\ skel/device (P(18), CPU, 0, 0) \\ skel/device (P(18), CPU, 0, 0) \\ skel/device (P(18), CPU, 0, 0) \\ skel/device (P(18), PU, 0, 0) \\ skel/device (P(18), P$
8		INFO:tensorflax:** Available Device: Devicettributes//jobiolablest/replica/bio INFO:tensorflax:** Available Device: Devicettributes//jobioalbox/replica/bio INFO:tensorflax:** Available Device: Devicettributes//jobiowrker/replica/bio INFO:tensorflax:** Available Device:***	//task//device/P016, CPU, 0, 0)           /task//device/P016, CPU, 0, 0)           sk:0/device/P016, CPU, 0, 0)
9		DFG-tensorTox:** Available Device: DevicAttributes(/jbiclaubest/replica/bit DFG-tensorTox:** Available Device: DevicAttributes(/jbiclaubest/replica/bit DFG-tensorTox:** Available Device: DevicAttributes(/jbiovarker/replica/bit DFG-tensorTox:** Available Device: DevicAttributes(/jbiovarker/replica/bit */	$\label{eq:constraint} \begin{array}{l} (T_{20}, k_{1}, 0) \in 0, 0 \\ (T_{20}, k_{1}, 0) \in (r_{10}, 0, 0) \in 0, 0 \\ (T_{20}, k_{1}, 0) \in (r_{10}, 0, 0) \\ (T_{20}, k_{1}, 0) \in (r_{10}, 0, 0) \\ (T_{20}, 0) \in (T_{20}, 0, 0) \\ (T_{20}, 0, 0, 0, 0, 0, 0) \\ (T_{20}, 0, 0, 0, 0, 0, 0, 0) \\ (T_{20}, 0, 0, 0, 0, 0, 0, 0, 0) \\ (T_{20}, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\ (T_{20}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\ (T_{20}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\ (T_{20}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\ (T_{20}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,$

Now, we can see total TPU cores are 8 and TPU workers = 1 and you can set it like how many workers you want you have seen in the previously how you can configure those things.

(Refer Slide Time: 09:39)

	-							
co	File	VIDHYA_XLA_TPU.ipynb Edit View Insert Runtime	© Tools Help <u>Last saved a</u>	101:11		Comment	Share 1	x 🚺
=	+ Cod	de + Text				Connect	• 🖌 Editi	ng 🔺
Q 0	0	INF0:tensorflow:*** Avai INF0:tensorflow:*** Avai INF0:tensorflow:*** Avai INF0:tensorflow:*** Avai Model: "sequential_1"	lable Device: Devic lable Device: Devic lable Device: Devic lable Device: Devic	eAttributes(/job:worker/replic eAttributes(/job:worker/replic eAttributes(/job:worker/replic eAttributes(/job:worker/replic	a:0/task:0/device:TPU_S a:0/task:0/device:TPU_S a:0/task:0/device:XLA_C a:0/task:0/device:XLA_C	YSTEM:0, ↑ ↓ YSTEM:0, TPU_ST PU:0, XLA_CPU, PU:0, XLA_CPU,	(0) 🖬 🏚 💭 (0, 0) (0, 0)	11
[1]		Layer (type)	Output Shape	Param #				
[4]		flatten 1 (Flatten)	(None, 784)	8				
D		- · · · · ·	(New 120)	100100				
		dense_2 (Dense)	(None, 128)	100480				
		dense_3 (Dense)	(None, 10)	1290				
		Total params: 101,770 Trainable params: 101,777 Non-trainable params: 0	9					
		time: 412 ms (started: 2	022-03-03 15:15:48 +	00:00)				
	11	# Train Size = 60000						
		# Batch Size = 200						
		# No. of Batches = 60000	//200 = 300					
_		# No. of Epochs = 2						
		# Tort Size = 10000						
		# Test 5126 - 10000						

(Refer Slide Time: 09:43)



And then model summary you have seen and then we are fitting the model to the model fitting is essentially running the model for you generated graph for that and you are using that and total of 16.8 seconds you are spending for your training of 2 epochs.

### (Refer Slide Time: 10:05)



Now, we are moving towards the example where we want to enable the XLA, but for with the TPU and also with tf.function runtime because the tf.function runtime has its own optimizations that you have seen. Because it is serializing the computational graph, that is not possible for your model.fit() and that is why the performance was not that good simple.

(Refer Slide Time: 10:32)



(Refer Slide Time: 10:38)

6 & VIDHYA\_XLA\_TPU.ipynb NPTEL ment 🔐 Share 🏚 🚺 Cor sert Runtime Tools Help Last saved at 01:11 0 0 File Edit View + Code + Text Connect 👻 🧪 Editing 🖍 tr.conig.experimental\_connect\_to\_cluster(resolver)
# This is the TPU initialization code that has to be at the beginning.
tf.tpu.experimental.initialize\_tpu\_system(resolver) 0 device\_name = tf.config.list\_logical\_devices('TPU')
print('Found TPU at: {}'.format(device\_name)) ? 7 INFO:tensorflow/Deallocate tpu buffers before initializing tpu system. INFO:tensorflow/Deallogate tpu buffers before initializing tpu system. INFO:tensorflow/DEallogate tpu buffers before initializing tpu system. MARDIME:tensorflow/TPU system groc://18.101.138.122:3470 has already been initialized. Reinitializing the TPU can cause previously c INFO:tensorflow/TPU system groc://10.101.138.122:3470 INFO:tensorflow/TPU system groc://10.101.138.122:3470 INFO:tensorflow/Teinished initializing TPU system INFO:tensorflow/Finished initializing TPU system. INFO: Ð • 9 [] # Data Required # REF: https://developers.googleblog.com/2017/03/xla-tensorflow-compiled.html - SOFTMAX FUNCTION
# REF: https://www.tensorflow.org/xla - XLA
# REF: https://www.tensorflow.org/xla/tutorials/jit\_compile - TUTORIAL # Eager execution is a powerful execution environment that evaluates operations immediately. # It does not build graphs, and the operations return actual values instead of computational graphs to run later.

(Refer Slide Time: 10:44)



Now, the same setup importing the TensorFlow all the data set libraries then initializing the resolvers right.

(Refer Slide Time: 10:48)

0	& VIDHYA_XLA_TPU.ipynb ☆	1	Comment	Share	¢ 🕥	N
+	File Edit View Insert Runtime Tools Help Last saved at 07:11 Code + Text		Connect	• /Fd	ting A	
=	redistribution_info=,					
Q						
	time: 1.8 ms (started: 2022-03-03 14:43:36 +00:00)					
0						
{x}	# viewing the train dataset					
0	<pre>df = tfds.as_dataframe(ds_train.take(5), ds_info) df</pre>					
	C+ image label					
	o 🖌 4					
	1 1					
	2 0					
	3 7 7					
	4 2 8					
	time: 1.21 s (started: 2022-03-03 14:43:38 +00:00)					
	[ ] # Viewing the Test dataset					
	df = tfds.as dataframe(ds test.take(5), ds info)					r

(Refer Slide Time: 10:53)

	v 006 0,> ± () :	*
COO € VIDHYA_XLA_TPU.jpynb ☆ File Edit View Insert Runtime Tools Help Last saved at 0111	🗖 Comment 🗮 Share 🏚 🚺	NPTE
= + Code + Text	Connect 👻 🧨 Editing 🔺	
time: 1.21 s (started: 2022-03-03 14:43:38 +00:00)		
# Viewing the Test dataset		
<pre>df = tfds.as_dataframe(ds_test.take(5), ds_info) (x) df</pre>		
□ C· image label 0 2 2		
1 D 0 2 <b>4</b>		
3 8 8		
4 7 7		
time: 1.21 s (started: 2022-03-03 14:43:42 +00:00)		
<pre>[] # Build a Training Pipeline # REF: https://www.tensorflow.org/datasets/keras_example</pre>		
# Only shuffle and repeat the dataset in training. The advantage of having an # infinite dataset for training is to avoid the potential last partial batch		
# in each enorth so that you don't need to think about scalion the oradients		20
		E.

And something regarding the data and setting up the just viewing the data frame also you can see these are the handwritten dates images from the data set itself.

(Refer Slide Time: 10:56)



And then setting up the input pipeline data input or data loading pipeline for normalizing mapping caching shuffling batching and prefetching.

(Refer Slide Time: 11:08)

		MIDLINA VIA TOULS										-	
co	) "	VIDHTA_XLA_IPU.Ipynb	и			Comment		s Si	hare	\$		V	
	File	e Edit View Insert Runtime	Tools Help Last saved at 0	1.11								-	
_	+ Co	ide + Text				Conne	ct •		18	diting	9	^	
-		us_test - us_test.mopt											
	11	normalize_img, num_p	arallel_calls=tf.data.	AUTOTUNE)									
		ds test = ds test.cache(	)										
		ds_test = ds_test.batch(	200)										
1		ds_test = ds_test.prefet	ch(tf.data.AUTOTUNE)										
-1		time, 50 mc (started, 30	22 02 02 14.59.07 .00.	00)									
*5		Line: 39 ms (started: 20	22-03-03 14:30:07 400:	00)									
										-	-		
_						$\uparrow$	4 o		\$	Ø	Î	:	
5	0	with tf.device('/TPU:0')	:			1	¢ a		\$	Ð	Î	:	
5	0	with tf.device('/TPU:0') model = tf.keras.model	: s.Sequential([	1)) # A Elattan lawar in Koras ras	hange the tensor t	^ .	↓ o		\$	0	1	:	
כ	0	<pre>with tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.Flatte tf.keras.layers.Dense(</pre>	: s.Sequential([ n(input_shape=(28, 28, 128,activation='relu',	.1)), # A Flatten layer in Keras res kernel regularizer≖tf.keras.regulari	hapes the tensor to zers.12(1=1e-4)), a	↑ o have a s # Dense in	hapi	e tha	¢ t i	equere ope	ual	: l to atio	
5	0	with tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.Flatte tf.keras.layers.Dense( tf.keras.layers.Dense(	: s.Sequential([ n(input_shape=(28, 28, 128,activation="relu", NUM_CLASSES, kernel_re	1)), # A Flatten layer in Keras res kernel_regularizer=tf.keras.regulari gularizer=tf.keras.regularizers.12(1	hapes the tensor to zers.12(l=le-4)), a =le-4))	↑ o have a s # Dense im	↓ c hap	e tha	¢ t i th	eques ope	ual	: l to atio	
5	0	with tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.Flatte tf.keras.layers.Dense( tf.keras.layers.Dense( ])	: s.Sequential([ n(input_shape=(28, 28, 128,activation='relu', NUM_CLASSES, kernel_re	.]), # A Flatten layer in Keras res kernel regularizer=tf.keras.regulari gularizer=tf.keras.regularizers.l2(1	hapes the tensor to zers.l2(l=le-4)), 4 =le-4))	↑ o have a s # Dense im	↓ o	e tha	¢ t i th	equer ope	ual	: l to atio	
5	0	<pre>with tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.Flatte tf.keras.layers.Dense( tf.keras.layers.Dense( ])</pre>	: s.Sequential{[ n(input_shape=(28, 28, 128,activation='relu', NUM_CLASSBS, kernel_re	1)), # A Flatten layer in Keras res kernel_regularizer=tf.keras.regulari gularizer=tf.keras.regularizers.l2(l	hapes the tensor to zers.l2(l=le-4)), = =le-4))	↑ o have a s # Dense in	↓ d hap	e tha	¢ t i	eques ope	aual era	: l to atio	
5	0	<pre>with tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.late tf.keras.layers.Dense( tf.keras.layers.Dense( ]) model.summary()</pre>	: s.Sequential{[ n(input_shape=(28, 28, 128,activation='relu', NUM_CLASSES, kernel_re	.]), # A Flatten layer in Keras res kernel_regularizer=tf.keras.regulari gularizer=tf.keras.regularizers.l2(l	hapes the tensor to zers.l2(l=le-4)), a =le-4))	↑ o have a s # Dense im	↓ d hap	e tha	¢ t i th	e equ	aual era	i l to atio	
5	0	<pre>with tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.Flatte tf.keras.layers.Dense( tf.keras.layers.Dense( ]) model.summary()</pre>	: s.Sequential([ n(input_shape=(28, 28, n(input_shape=relut, num_cLASSES, kernel_re	. 1)), # A Flatten løyer in Koras res kornel regularizer=tf.koras.regulari gularizer=tf.koras.regularizers.l2(L	hapes the tensor t zers.l2(l=le-4)), = =le-4))	↑ o have a s # Dense im	↓ c	e tha	¢ it i i th	eque ope	aual era	: l to atio	
	0	<pre>vith tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.Flatte tf.keras.layers.Bense( tf.keras.layers.Dense( l] model.summary() Hodel: "sequential_15"</pre>	: s.Sequential([ n(input_shape=(28, 28, 128,activgtion='relu', NUM_CLASSES, kernel_re	.)), # A Flatten layer in Keras res kenel regularizer=tf.keras.regulari gularizer=tf.keras.regularizers.12(1	hapes the tensor to zers.l2(l=le-4)), d =le-4))	 o have a s # Dense im	↓ c	e tha	¢ it i i th	s equ	aual era	: L to atio	
2	0	<pre>vith tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.Flatte tf.keras.layers.Bense( 1) model.summary() Model: "sequential_15" Layer (type)</pre>	: s.Sequential{[ n(input_shape=(28, 28, 128,activgtion='relu', NUM_CLASSBS, kernel_re Output_Shape	1)), # A Flatten layer in Kerss res kenel regularizeretf.keras.regulari gularizer=tf.keras.regularizers.12(1 Paran #	hapes the tensor to zers.l2(l=le-4)), + =le-4))	 o have a s # Dense im	↓ c	e tha	¢ th	ر s equ s op	ual	: L to atio	
5	0	<pre>vith tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.Flatte tf.keras.layers.Dense(</pre>	: s.Sequential([ n(input shape=(28, 28, 128,activgtion=relu", NUM CLASSES, kernel_re Output Shape (None, 784)	.1)), # A Flatten layer in Kerss res (senel_regularizer-tf.keras.regulari gularizer-tf.keras.regularizers.12(1 Paran # 0	hapes the tensor to zers.l2(t=l=-4)), i =l=-4))	 o have a s # Dense im	↓ c	e the	¢ t i th	D s equ s op	ual era	: l to atio	
	D	<pre>vith tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.Flatte tf.keras.layers.Bense( ]) model.summary() Model: "sequential_15" Layer (type) flatten_15 (flatten)</pre>	: s.Sequential([ n(input_shape=(2%, 28, 128,activgtion=relut, 128,activgtion=relut, NUM_CLASSES, kernel_re Output_Shape (None, 784)	)), # A Flatten layer in Keras res kenel regularizer=tf.keras.regulari gularizer=tf.keras.regularizers.l2(l Paran # 0	hapes the tensor to zers.l2(t=ie-4)), = =ie-4))	↑ Ø have a s Ø Dense in	↓ c	e tha	¢ t 1 th	E s equ	ual	: l to atio	
	D	<pre>vith tf.device('/TPU:0') model = tf.keras.model tf.keras.layers.Platte tf.keras.layers.Dense( )) model.summary() Model: "sequential_15" Layer (type) flatten_15 (Flatten) dense_30 (Dense)</pre>	: 5.Sequential{[ 1(Input shape<[2, 8, 2] 2(2),activation=relu', NUM_CLASSES, kernel_re Output Shape (None, 784) (None, 128)	1)), # A Flatten layer in Kerss res kernel regularizer=tf.kerss.regulari gularizer=tf.kerss.regularizers.l2(1 Param # 0 100480	hapes the tensor to rers.l2(tele-4)), i -le-4))	↑ Ø have a s Ø Dense in	↓ c	e tha	¢ t i th	E s equ	a ual era	: l to atio	

And then with this TPU device. So, one TPU device we are using again you can also use any other strategy also. So, sequential model that we have defined for that device. (Refer Slide Time: 11:22)



And then defining the optimizer and loss function. But now we are actually defining the training step, but not using the model of fit.

(Refer Slide Time: 11:41)



And we are wrapping it with tf dot function runtime with JIT compiler equal to True and after doing that we are actually calling the train step for these number same number of epochs. Because we have run for two epochs previously and 4 times speedup that we talked about in the session in the slides you can see 4 times speedup you are getting right.

So, this is a very simple example where you can set up your own pipeline for your training and you can define like, which exactly steps which are the steps that will be wrapped with your with your trained tf dot function runtime which JIT compiler. And this is one example where we have done in TPU ok.

Now, we can extend this for running it several TPUs and see what are the strategies supported for a XLA compile and play with it ok. So, this is where your exploration for different strategies and you know whatever methods that you have learned from this course we will apply right. So, you take this code and play with it right.

You can also use keras data set and see what happens for the particular example that we have seen. Because we have we have used tfds the TensorFlow data set and try to use keras data set and see what happens ok.

(Refer Slide Time: 13:28)



So, explore all these things you can note it down and you can explore this, because a lot more interesting things that you can discover I am telling you ok. So, let us go into the GPU now right. So, we have seen the TPU part where we have enabled the TPU execution setup environment and wrapped the function with the tf.function(jit\_compile = True).

Now we will train first one naive way because we want to see right what is happening here with the GPU and of course, without amp the automatic mixed precision without XLA without anything its vanilla training model right.

(Refer Slide Time: 14:09)



So, where you have the strengths of imported the image the same thing that you have seen in the TensorFlow model.

(Refer Slide Time: 14:28)



So, same training module here train set test data set we are unpacking it model defining with the sequential api model summary we are checking. And then clearing the session if we have, let us say running previous JIT compilation session is already on we are just doing it just for safety. We are clearing the session and setting the XLA disable model dot compile will compile the model train model with this fit function right. So, that is it that is the vanilla training for the TensorFlow particular pattern.



(Refer Slide Time: 14:57)

So, interestingly your DLProf will figure it out that you are actually not using XLA think I do not know. So, we will see I hope this is the profile for after training with this.

(Refer Slide Time: 15:07)



So, basically you can see if amp was not a enabled GPU was not greatly used and so on and so forth. Unable to split profile into training iterations, so ok that is ok. GPU memory is underutilized that is fine because of the very small model right.



(Refer Slide Time: 15:34)

And all the operations that are being actually called and executed how many times of operations execution call and how much time you have spent for those operation call that you can see profile it here.

(Refer Slide Time: 15:51)



And you can see few of the operations that are matrix multiplication operations which are actually tensor core eligible or not eligible. So, whether using tensor core now we are not using because we are mixed precision.

And all these profiles you can get it from right. So, ok so just to see like this is very my way of setting things up right.

(Refer Slide Time: 16:24)



Now, of course, average time you can see 12.8 now of course, we want it accelerated right. So, let us see what happens if we enable the accelerate right.

(Refer Slide Time: 16:34)



(Refer Slide Time: 16:41)



### (Refer Slide Time: 16:46)



So, the same thing same train the data set unpacking model definitions are clearing in the background and here we are actually enabling the XLA and then model compile model train model with fit function and train model. So, this is a simple same just with two lines of code we are actually enabling the XLA.

# (Refer Slide Time: 17:05)



## (Refer Slide Time: 17:16)

NVIDIA D	LProf Viewer	View Dashboard -				
				More	3,055,145 AssignAdd/tariableOp_1	P AssignAddVterie P
					2.921.338 Cast_4	Cast
System Config		Expert Systems		Guidance		
GPU Count	1	Problem	Recommendation	Understanding GPU util	zation and timing details of the operations is the fir	st step in
GPU Name(s)	Quadro RTX 5000	W Unable to determine if AMP	If AMP wasn't enabled, try enabling it. For	To learn more al	bout Tensor cores and Mixed Precision training, vis	it this
CPU Model	Intel(R) Xeon(R) W-2255 CPU @	(Automatic Mixed Precision) was enabled	more information: https://developer.nvidia. com/automatic-mixed-precision	site https://devel • You will find reso full use of Tenso	oper midia comhensor_cones surces on how to train networks with mixed precisi r cores for Tensorflow models here.	on and make
GPU Driver Version	3.70GHz 470.86	The GPU is underutilized: Only 2.9% of the profiled time is spent on GPU kernel	"Other" has the highest (non-GPU) usage at 57.4%. Investigate framework and system overhead	https://docs.mid training/index.ht Note that if there	la com/deepleaning/solchised precision- ni#traning_tensorflow e are multiple komets being observed on single op.	these are
Framework	TensorFlow 2.6.0	operations		likely performing	data transposes to prepare the data for efficient u	se by
CUDA Version	11.5	Unable to split profile into	Specify key op by setting thekey_op	tensorcores, su	un elemposes memoerves would not use tensor co	res.
cuDNN Version	8.3.0	found	aynes			
NSys Version	2021.3.2.4- 0275341	GPU Memory is underutilized. Only 7% of GPU Memory is	Try increasing batch size by 4x to increase data throughput			
DLProf CLI Version	v1.7.0/r21.11	used				
DLProf DB Version	1.7.0					
DLPtof Viewer Version	170/2111					

So, let us see what happened here average time got reduced to 7.75 that is good and, but still. So, if you see of course, amp was not used and GPU is underutilized that is understandable.

### (Refer Slide Time: 17:25)

NVIDIA DLProf Viewer View Dashboard •							(ð Help 🔹
Using Tensor Cores Not Using Tensor Cores	ther	GPU Time (ns)	Op Name	Ор Туре	Calls	TC Eligible	Using TC
Outside Aggregation		132,840,516	duster_0_1/kla_run	_XiaRun	1200	~	x
22 2 10		82,157,596	cluster_0_1/kla_compile	_XlaCompile	1200	~	x
lojte 0.5		22,036,920	unassociated_cuMemopyDto HAsymc_v2	cuMemopyOtoHAsyn c_v2	2418	×	x
		11,929,216	IteratorGetNext/_1	_Send	1400	×	×
0		11,734,690	cluster_2_1/kla_run	_XlaRun	200	×	x
Iterations	5.744,371	unassociated_cuMemcpyHto DAsymc_v2	cuMemcpyHtoDAsyn c_V2	33	×	×	
Zoom: enabled (7.65x), tooltip: enabled	5,215,074	cluster_1_1/kia_run	_XiaRun	1200	×	x	
Reset zoom Toggle Tooltip	More	3,108,986	AssignAddVariableOp_3	AssignAdd/tariebleO p	1400	×	x
		3,065,145	AssignAddVariableOp_1	AssignAddVariableO p	1400	×	x
		2,921,338	Cast_4	Cast	1400	×	x
ert Systems	Guidance						
Problem Recommendation	Understanding GPU utila profiling your model.	cation and timing detail	s of the operations is the first st	ep in			
Unable to determine if AMP If AMP wasn't enabled, try enabling it. For (Automatic Mixed Precision) more information: https://developer.midia.	To learn the about Tentor cores and Mixed Precision training, visit this     she tittps://developer.nvdia.com/tenzor_cores						

And of course, again you can see the functions, that is and also you can see which is the operation that is being actually run by your optimized code ok. So, for the from the XLA part.

I have not shown other views you can exploit those. So, as I was mentioning that it is not possible every time to go to every tab and explain you everything, but you can explore the other views as well.

(Refer Slide Time: 17:59)

WIDIA D	LProf Viewer	View Ops an	nd Kernels •						
	(		R						
Ops and R	lerner Sl	ummaries	6						
¥ Ops									
O Click an op t	o see its kernels	Delow							
Show 10 v entries			Export to: Excel PDF	Clipboard CSV	JSON			Search:	
GPU Time (ns) 🔻	CPU Time (ns)	Op ID	Op Name	Ор Туре	Calls +	TC Eligible	Using TC	Kernel Calls	Data Type
Search GPL	Search CPU	Search Op II	Search Op Name	Search Op 1	Search C.	0 or 1	0 or 1	Search Ke	Search D
120,518,760	1,237,654,102	INFERENCE_T RAIN_STEP_898_ 1	_inference_train_step_898	interence_train_ step_898	600	×	x	14,560	
32,259,186	39,220,555	CUMEMCPYHTD DASYNC_V2_1	unassociated_cuMemopyHtoDAs ync_v2	cuMemcpyHtoDAs ync_v2	1225	×	×	0	
12,444,404	11,474,382	BIASADD_1	a_inference_train_step_898_X0a MustCompile_true_config_proto_ n_007_n_082_001_000_ex ecutor_type See more ♥	BiasAdd	642	×	×	618	
			a inference train step 898 Xia						

Let us say you want to see the operations and kernels that are being executed you can see the list of the operations and all the statistics behind those you can export them in PDF, CSV, JSON format or excel format whatever you want. Kernel's by operations iteration kernels by operation iterations.

(Refer Slide Time: 18:15)

		erations					onep v
Iteration	S						
Durations Acro	ss Iterations						
		To res	aggregate network, chang	e values and click button		Total = 1 Profiled = 1 Sta	rt=0
		Select Iter Start 0	* Select Iter S	op 0 🛟		Stop = 0	
		Select Key Node		1 Reago	(class)		
Usi	g Tensor Cores Not Using Tenso	r Cores Memory	Dataloader	IO CPU	Other Outsid	e Aggregation New Ph	ofile Boundary
14							
12	1						
10							
(2)							
ration (s) co							

So, all the things you can explore from this profiler right. But of course, this is where you will explore more ok.

(Refer Slide Time: 18:26)



Now we want to introduce amp also and enable XLA at the same time right.

Now, we want more performance we are just we do not know what is what will happen right, but we are just we know that if we enable amp and use mixed precision then we can you know use the tensor cores more ok we can increase the utilization, so let us do that.

And also let us keep the XLA on right. So, I do not know what will happen. So, let us see eager execution. So, enabling the eager execution here the image size all the images train test data set unpacking and definition of the model, model work summary.

(Refer Slide Time: 19:11)



(Refer Slide Time: 19:16)



### (Refer Slide Time: 19:18)



And here we are enabling XLA also at the same time. So, in the previous example in the prior class that you have seen, how we are getting more utilized tensor cores by using mixed precision in our pytorch class.

So, extensively we have not used mixed precision in TensorFlow, but you can use that how you can use that here with optimizer basically you can scale it right. So, experimental dot enable mixed precision graph rewrite you can enable that for this optimizer and you can use that optimizer inside your model and train that model as simple as that use mixed precision.

### (Refer Slide Time: 20:07)



And let us see what happened here and oops our average iteration time got increased so; that means, we have degraded performance. So that means, the, but introducing everything XLA mixed precision everything in the tensor processing unit which is the GPU we were getting much more performance right and that is because the that that is because Google ok.

(Refer Slide Time: 20:47)



### (Refer Slide Time: 20:50)



So, basically google supports the mixed precision binary fusion which is supported in GPU, but not supported inside your GPU and you will not keep tensor course utilized. So, still your GPU is underutilized and whatever XLA run you have defined it is not using the tensor core.

But some of the in the cluster of 5 a xla run this 200 is using the tensor core ok. So, some parts will not be actually compatible for your target gpu. So, you just need to use some libraries.

So, now here we are using the amp library from Google itself again you know there are two libraries for that one is NVIDIA apex and the native library from PyTorch both from PyTorch and the tensor core. Now you can know what is what we use for getting enhanced performance right. (Refer Slide Time: 21:43)

🗂 jupyt	ter train_xla_amp_tfds_custom.py+ 17 hours ago	Logout
File Ed	t View Language	Python
1 # Impo	ort Necessary libraries	-
2 3 import	t tensorflow as tf	
4 import	t tensorflow_datasets as tfds	
5 5 # Data	a Renul red	
7		
8 # REF:	: https://developers.googleblog.com/2017/03/xla-Tensorflow-compiled.html - SOFTMAX FUNCTION	
9 # REF: 0 # REF:	: https://www.tensoritow.org/xla - XLM : https://www.tensorflow.org/xla/tutorials/jit compile - TUTORIAL	
1		
2 # Eage 3 # It (	er execution is a powerful execution environment that evaluates operations immediately. does not build graphs, and the operations return actual values instead of computational graphs to run later.	
4		
5 # Prep	paring Data	
7 # Size	e of each input image, 28 x 28 pixels	
8 IMAGE	SIZE = 28 * 28	
B NUM CL	ASSES = 10	
1		
12 # Load	ds MNIST dataset.	
4 (ds_tr	rain, ds_test), ds_info = tfds.load('mnist', split=['train', 'test'], shuffle_files=True,as_supervised = True, w	ith_info =
True, t	try_gcs = True) #try_gcs for TPU	
26 # Buil	ld a Training Pipeline	1 Mar 1997
27 # REF:	: https://www.tensorflow.org/datasets/keras_example	
28		

Now in the next setting, we will see how you can actually also enhance the training pipeline with tfds right. So, we will now formulate the pipeline with tfds right as you have seen in the TPUs right.

(Refer Slide Time: 22:04)



(Refer Slide Time: 22:14)



So, training data with tfds dot load from mnist splitting training test and comparing and formulating the pipeline here.

(Refer Slide Time: 22:24)



And then map cache batch prefetch sequential model definition model summary optimizer and mixed precision enabled sparse category loss we are using here from the logics, that we are getting from the last layer and defining the train step with runtime function tf.function(jit\_compile = True) with this we have wrapped this function.

And. So, I do not know everything we are using here. So, the tfds data pipeline the JIT compiler for the function training step and mixed precision caching of data load I mean I do not know what else we can have here right.



(Refer Slide Time: 23:05)

So, let us see the dlprof ok. So, with enabling XLA we will not give you always the best performance that I have told you in the slides right. So, that is exactly what we are doing here 12.8 it is not exactly what we want. So, if you blindly just enable everything that does not mean that you will get the better performance.

Conclusion is you see analyze what is best for you and actually what you will get from the exploration and profiling. So, use the profiler always like how you can actually see where is the performance you are going to increase or decrease depending on which strategy you are using. So, all these are basically exploration based ok.

And for your target GPUs for your target TPUs what will be the best for you explore that and see what actually fits for you ok. What are the optimizations that are good for your target architecture you explore. You explore this the Kernel's here that is that that are being generated and see what kind of visions you have after the training.

You can inline the you can inline to see the intermediate results. So, I have shown you in the class slide that part of the techniques for enabling the optimizer graph for generated. So, you can use this and see what kind of optimized graph you see. So, with this we will conclude the session and.

Thank you for that.