TensorFlow Developer Certificate in 2022: Zero To Mastery

What is Deep Learning

- Type of machine learning based on artificial neural networks in which multiple layers of processing are used to extract progressively higher level features from data
- Machine Learning is turning data into numbers and finding patterns in those numbers
- With Machine learning start with the input and ideal output
  - Then show examples of the output and the algorithm will figure out the ideal rules to get form the input to the output

Why Use Deep Learning

- Reason : For a complex problem, can you think of all the rules
- If you can build a simple rule-based system that does not require machine learning do that
- What is deep learning good for
  - Problems with long list of rules
  - Continually changing environments – Can adapt to new scenarios
  - Discovering insights within large collections of data
- What is Deep Learning not good for
  - When you need explainability – the patterns learned by a deep learning model are typically uninterpretable by a human
  - When the traditional approach is a better option
  - When errors are unacceptable – Outputs of learning model are not always predictable
  - When you don’t have much data – Require large amount of data to produce great results (Well see how to get great result without much data)
- Machine learning work best with structured data and deep learning performs better on unstructured data (Natural Language Processing, Sound Waves)

What are Neural Networks

- Steps
  - Inputs – Could be pictures, tweet, sound wave or anything else
    - Numerical Encoding – Convert the inputs into numbers and put them into the neural network
  - Learns representation (Patterns/features/weights)
    - finds a pattern in the numbers
  - Representation Outputs – transcribe the Outputs
    - Representation Outputs are the patterns that have been found.
  - Outputs – We convert them into output a human can use.
- Anatomy of a neural network
  - Input layer
  - hidden layers (learn patterns in data)
    - learn the patterns between the input features and the output
  - Output Layer – Outputs learned representation or prediction probabilities
- Patterns is an arbitrary term you will often hear “embedding”, weights, features representation and feature vectors

What are Networks Networks

- Type of learning
  - Supervised Learning — Uses data and labels (ex. images of food and text associated with images)
  - Semi-supervised Learning — Only some data has labels – (train on labeled items)
  - Unsupervised Learning — Only have data and find the patterns.
  - Transfer Learning — Taking what one deep learning model has learned and use it on another problem

What is Deep Learning Used for

- Use Cases
  - Recommendation
  - Translation — Sequence to Sequence (Translate a set of words into another set)
  - Speech Recognition
  - Computer Vision
  - Natural Language Processing
  - Classification or Regression — Is it one thing or another
  - Regression — independent variables produce a target variable
  - Protein Folding

What is and why use TensorFlow

- End to End platform for machine learning
- Write fast deep learning code in Python/other accessible languages (able to run on GPU/TPU (Tensor Processing Unit))
- Able to access many pre-built deep learning model (TensorFlow Hub) – Transfer Learning
- Whole Stack: Preprocess data, model data (build a neural network) , deploy mode in your application
- Why Tensor Flow
  - Easy model building using intuitive high level APIs
  - Robust ML production anywhere. Train and deploy models in the cloud, on-premises or in the browser or devices.
  - Powerful Experimentation for research
- With tools like Colab, Keras, TensorFlow virtually anyone can solve in a day with no initial investment problems that would have required an engineering team working for a quarter and $20K in 2014
- What is a GPU/TPU
  - GPU – Better at finding patterns in numbers
  - TPU – AI accelerator application specific integrated developed specifically for neural network machine learning

What is a Tensor

- A numerical way to represent information usually the input and the output
- Tensor Flow (flow of the tensor)
  - Inputs
  - Numerical Encoding
  - Learns (representation (patterns/features/weights))
  - Representation outputs
  - Outputs
- Primary data structure used by tensor
  - input/output/transformations are represented as tensors

Using the @tf.function

- Turns a Python function into a callable Tensorflow graph
  - Must decorate it with @tf.function
  - Converts it into a faster version of itself by making it part of a computation graph
- See https://www.tensorflow.org/guide/function
- A transformation tool that creates Python Independent dataflow graphs out of your Python Code
  - Help to create performant and portable models and it is required to use SavedModel
- Recommendations
  - Debug in eager mode then decorate with @tf.function
  - Don’t rely on on Python side effect like object mutation or list appends
  - tf.function works best with TensorFlow ops
    - NumPy and Python calls are best converted to constants

Make sure our tensor operations run really fast on a gpu

- A tensor flow tensor can be run on a GPU for faster presenting
- GPU are really fast at numerical compting
- TPU – An AI Accelerator application specific integrated circuit developed for neural network machine learning for Tensorflow
- How might you use the GPU
  - tf.config.list_physical_devices() // Get a list of devices on your machine
  - tf.config.list_physical_devices(“GPU”) // Get list of GPU only
- Colab can provide free access to a GPU – Can we do this in Pycharm
- nvidia-smi – // Gives the description of the GPU
- If you have access to a cuda gpu then tensorflow will automatically use it.
- For the example there was no difference between the code and the

Introduction to Neural Network Regression with TensorFlow

- Regression : Predicting a number
  - ex. Regression Problems
    - predict coordinates of the boxes for an object detection problem
    - How much will the house sell for.
    - predict the cost of medical insurance for an individual given their demographics (age, sex gender, race)
  - Tell a regression Problem – How much, How many
- In Machine Learning and Deep Learning - how you define the problem will determine how you approach a problem
- Notes
  - Dependent Variables is the outcome we are trying to predict
  - Independent variables – The features as a number
    - Example – The number of bedrooms In a house
    - Example – The size of the house
  - Regression Analysis – A statistical process for estimating the relationships between a dependent variable and one more independent variables

Inputs and outputs of a neural network regression problems

- Predict the sales price of the house
  - Inputs : # of bedrooms, #of bathrooms, # of Garages
  - How do we encode the inputs ? ex. (4 bedrooms, 2 bathtrooms, 2 garages)
    - Could use Hot Encoding – [[0,0,0,1], [0,1,0,0], [0,1,0,0]] 1-based to encode the input features
  - Output – The prediction
    - Comparing the predicted to the actual make the prediction better known as supervised learning
  - The shape should be [3] and the output is going to be [1] (For regression the output is one value)
    - Question : Why is the input shape 3 and [3,1,4]
      - Answer 3 input vectors – 1 for bedroom, 1 for bathroom, 1 for garage
- Looks at many examples and Input features and output prices it will learn the relationship between them
- Often an algorithm exist that is similar to your problem

Anatomy and Architecture of a neural network Regression Model

- Hyperparameter - Setting that can be changed. (A dial on your neural network to see if it improves)
- Hyperparameters for Regression
  - Input Layer Shape — Same shape as number of features (3 (bedrooms, bathrooms, car)
  - Hidden Layers — minimum 1, maximum = unlimited
  - Neurons per hidden layerProblem specific generally 10 to 100
  - Output Layer Shape — Same shape as desired prediction shape (1 for house price)
  - Hidden Activation — Usually ReLU (rectified linear Unit)
  - Output Activation — None, ReLU, logistic/tanh
  - Loss Function — MSE (mean square error) or MAE (mean absolute error) Huber (combination of MAE/MSE) if outliers
  - Optimizer — SGD (stochastic gradient descent), Adam
    - Improves the predictions
  - A hyperparameter in machine learning is something a data analyst or devel can set themselves
  - A Parameter describes something a model learns on its won (a value not explicity set by an analyst)

Anatomy and Architecture of a neural network Regression Model

- Tensorflow Code
  - model = tf.keras.Sequential({ // Create the model
  - tf.keras.Input(shape=-(3,)),
  - tf.keras.layers.Dense(100, activation=”relu”), // Contains neurons per hidden layer, activation parameter
  - tf.keras.layers.Dense(100, activation=”relu”), // 100 is the number of neurons
  - tf.keras.layers.Dense(100, activation=”relu”),
  - tf.keras.layers.Dense(1, activation=None)]) // Output
  - // Compile the model
- model.compile(
  - loss=tf.keras.losses.mae, // How wrong the relationships are
  - optimizer = tf.keras.optimizers.Adam(lr-0.0001), // How to improve the patterns to reduce the lost function
  - metrics[“mae”])
- // Fit the Model
- model.fit(x_train, y_train, epochs=100) // Train

Regression Input and Output Shapes Steps in Modeling Tensorflow

- The input shape is the shape of your data that goes into the model.
- The output shape is the shape of your data you want to come out of your model.
- Steps in Modeling Tensor Flow
  - 1. Create a model
    - define the input and output layers as wells as the hidden layers of a deep learning model
      - Can be a functional or Sequential API
    -
Can import a pretrained model relevant to your problem
  - 2. Compiling a model
  - Loss -- how wrong your models's predictions are compared to the truth labels (you want to minimize this)
  - Optimizer -- How your model should update its internal patterns to better make predictions
  - Metrics -- Human Interpretable values for how well your model is doing.
  - 3. Fitting a model
  - letting the model try to find patterns between X & y (features and labels)
  - Fit the model to the training data so it can discover patterns
  - Epochs -- how many times the model will go through all of the training sample
  - 4. Evaluate the model on the testing data (how reliable are our model's predictions)
  - Common Ways to improve a deep model

Introduction to neural network

- Examples
  - Is the email spam or not spam
    - Is it one thing or another
    - binary classification
  - Is the photo of sushi,k steak or pizza
    - Multi-class classification
  - What Tags should this article have
    - Mutlilabel classification
    - multiple label options per sample.

What is Tensorflow Developer Certification
Why the Tensor Flow Certification
How to prepare (your brain) for the Tensorflow Developer Certification

- Can you build a tensorflow model for
  - Regression
  - Image Classification
  - NLP Classification
  - Time Series Forecasting
- Given a certain dataset can you access what problem that data is trying to solve and then preprocess and build the model
- Why the Tensor Flow Certification
  - Test your skills and Show Case skills and Tensorflow Certicate Network
- How to prepare (your brain) for the Tensorflow Developer Certification
  - Make sure you go through the Welcome to Mastery TensorFlow for Deep Learning Book
  - Read the TensorFlow Developer Certification
    - Create a notebook and map the skills checklist with running code examples

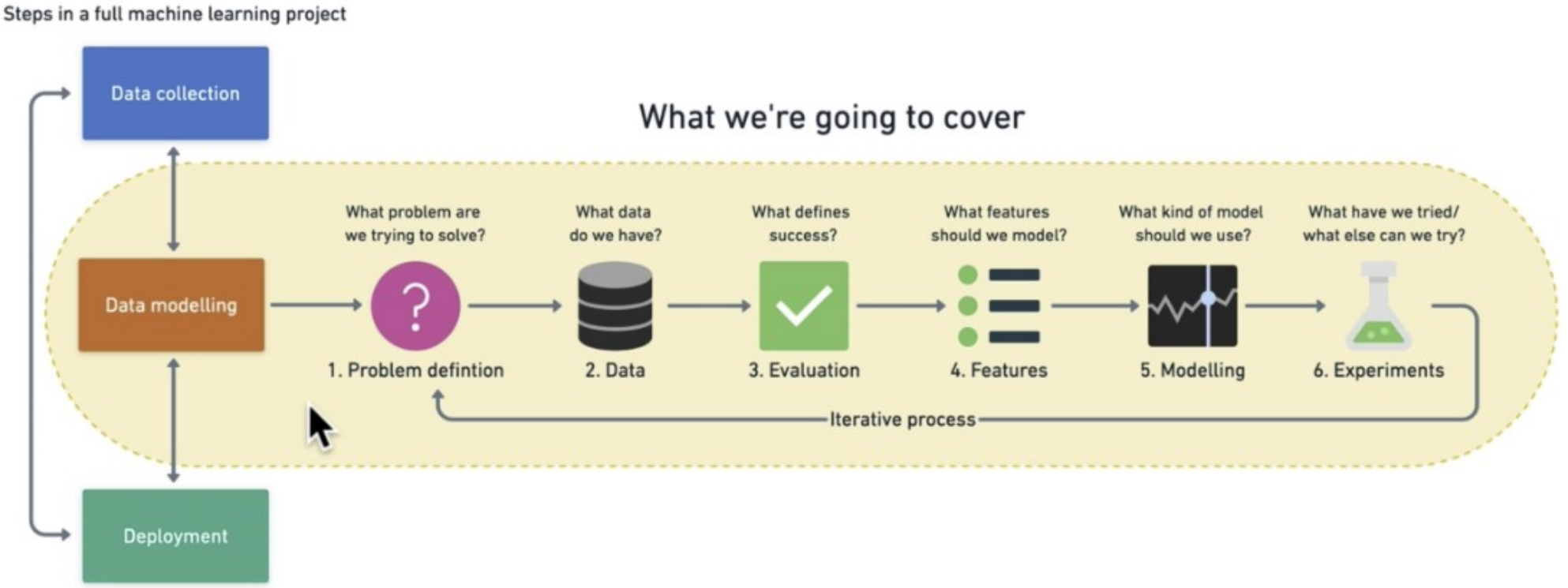
How to Compare your computer for TensorFlow Certification

- Read the setup environment to take the Tensorflow Develop Certificate Exam
- Go through PyCharm Tutorials
- Make sure Tensor Flow Pycharm
  - Train the model and save it to .h5 format
  - When submitted it will look for a directory
- If example model script runs in under 5 to 10 minutes in PyCharm your local machine should be good to go.
  - Found in the github image_classification_test.py
  - If the script takes longer than 5 minutes
    - Run the code in COLAB and put the model in the correct space for saving
- Datasets are not too large
- Troubleshooting Tidbits
  - .Input and Output Shapes – print these out if your stuck
  - Input and Output Datatypes – TensorFlow usually prefers float32
  - Output Activation Functions – for classification – sigmoid (binary classification) vs softmax (mutli class classification) which one should you use
  - Loss Functions – For Classification – sparse_categorical_crossentropy vs categorical_crossentropy which should be used
  - Way to improve the model –

Machine Learning Primer
AI/Machine Learning, DataScience

- The harder tasks are to describe the harder it is to tell machines what to do.
- With large amount of data, we can give the data to machine and allow them to make decisions.
- AI/Machine Learning/Data Science
  - Starts with AI
    - narrow AI Machines can be just as good or better than humans at a specific task
    - General AI
  - Machine Learning Subset of AI
    - An approach to achieve intelligence by finding patterns in a set of data
    - Have machines perform task or find data without being explicitly programmed
  - Deep Learning – A subset of Machine Learning
  - Data Science
    - Analyzing Data and then doing something with it.

How Did We Get Here



How did we get here
Types of Machine Learning

- Hardest part is grabbing the data. We need to find it and understand it.
  - Some data is noisy and messy. We must clean in it.
- Type of Machine Learning
  - Supervised
    - Classification — Is it an apple or pear
    - Regression — Predicting stock prices
  - Unsupervised
    - Clustering — Create groups of similar data
    - Association Rule Learning — Associate different things to predict what a customer might buy
  - Reinforcement
    - Skill Acquisition — Play a game and get a higher
    - Real Time Learning

What is Machine Learning

- Data + Machine Algorithm learning different patterns in the data and then use the algorithm (including the data it learned) and make predictions about the future
- Models – Machine Learning Algorithms with the data (patterns)
- Data Science – Run experiments on data with the hope of finding insights.
  - Data Analysis – Looking at set of data and gaining an understanding by looking at different examples
    - visualizations
  - Data Science – Running experiments on a set of data with the hope of finding actionable insights within it.
    - Machine Learning

Machine Learning – 6 Steps for Machine Learning

- See the Image a few slides above
- Steps
  - Problem Definitions
    - What kind of problem is it : Supervised, Unsupervised, Classification or Regression
  - Data
    - What kind of data do we have (Structured, Unstructured (Images, Audio))
  - Evaluation
    - What success means to us
    - What is the percentage accurate model that we need ?
  - Features
    - What do we know about the data (types Categorical and Derived)
  - Modeling
    - What machine algorithm should we use
    - Algorithms have already been coded.
    - Focus : Figure out the right model for the right kind of problem.
  - Experimentation
    - How can we improve / What to try next

Types of Machine Learning Problems

Problem Definition

- When shouldn't you use machine learning – Will a simple hand-code instruction baed system work
- Main Types of Machine Learning
  - Supervised Learning
    - Classification Is it one thing or another
      - Binary Classification – Two Options
      - Multi-Classification – more than two options
    - Regression – A Continuous Number – ex How many people will buy this app.
  - Unsupervised Learning
    - Example
      - The purchase history of all customers at your store and your marketing teams want to send out a promotion – Who is interested in it.
    - You apply the label to the data programatically
    - Called Clustering – Putting groups of similar examples together
  - Transfer Learning
    - Find patterns in Data then Machine Learning must make millions of calculations
    - Example – Finding objects in a picture can be reused (What Tree look like, grass)
  - Reinforcement Learning
    - Have a computer program perform actions in a defined space and reward/punish
    - Example : Chess, Asteroids
    - Yet to find its way into many applications

Types of Machine Learning Problems

- Align the problem you are trying to solve to a machine learning problem
  - Supervised Learning – Know your inputs and output
  - Unsupervised Learning – Inot ot sure of the outputs, but I have inputs. Trying to group them in some way
  - Transfer Learning – Can I leverage an older machine learning model.

Types of Data

- The more data you have the better chances to find more patterns
- What kind of data do we have
  - Structured
  - Unstructured
  - Static / Streaming
    - Static data does not change over time.
      - Usually you want a lot of these examples
    - Streaming
      - Data constantly over time.
      - Example. New Headlines
- Data Science Workflow
    - Open a CSV File in a Ipython Notebook
    - Use pandas for data analysis and visualizations from mathplotlib
    - Build machine learning models using SciKit

Types of Evaluation Features of Data

- A measure of how well a machine learning algorithm predicts the future
- Difference Evaluation Metrics
  - Classification Regression Recommendation
  - Accuracy Mean Absolute Error (MAE) Precision at K (Only care about the top N)
  - Precision Mean Squared Error (MSE)
  - Recall Root Mean Squared Error (RMSE)
- Features of Data
  - What do we already know about data
  - Features – Different forms of data within Structured/Unstructured Data
    - Feature Variables – Features of the data
    - Used to predict the target variables
    - Different Kinds
      - Numerical Features
      - Categorical Features
      - Derived – Creates a new feature using the existing ones
        - Feature Engineering – Looking at different features of data and creating new ones/altering existing one
    - Unstructured Data – Has features, but are less obvious
    - A feature works best in machine learning algorithm if many of the samples have it
  - Feature Coverage – How many sample have different features per column
  - Want at least 10% Coverage at least

Splitting Data

- 3 parts to modeling
  - Choosing and training a model
  - Tuning the model
  - Model Comparison
- Based on problem and data what model should we use
- Most important concept in machine learning – Training, Validation and test Sets
- Your Data should be split into
  - Training — Train your model on this — usually 70 to 80 %
  - Validation — Tune your model on this — usually 10 to 15 %
  - Test — Test and compare on this — usually 10 to 15 %
- Generalization – Ability for a machine learning model to perform well on data it has not seen before.

Picking the Model

- Working with Structured Data
  - Decision Trees
  - Random Forest
- Unstructured
  - Deep Learning
  - Neural Networks
  - Transfer Learning
- Train the model by lining up the inputs with the outputs
  - Biggest goal is to minimize time between experiments
    - Use a small amount of data first to decrease the time

Tuning the Model

- Many model have different hyper parameters that can be adjusted.
  - Example - Random Forest will adjust the number of trees
  - Example – Neural Network will allow the adjustment of the hidden layers
- Tuning can take place on training or validation data sets
- Machine learning models have hyperparameters you can adjust

Model Comparison

- How will a model perform in the real world
- Using your test set it will show how you model generalizes (gets the values right even if it never was tested/validated with those inputs)
  - A good model will have the same performance on the training and test set and validation
    - Not uncommon to see a slight decline in performance from the model
  - not uncomment to see a degradation in conformance on test set
    - Should be worried about
      - Underfitting Training set is higher than test set
      - Overfitting Test Set is higher than training set
        - learn the patterns too well in the dataset
  - Overfitting and Underfitting are examples of a model not be able to generalize well
  - Main reason for Underfitting and Overfitting
    - When the Training Data leaks into Test Data results in overfitting
    - Data Mismatch – Data is different (having different features) then the test set results in underfitting
  - Fixing for underfitting
    - Try a more advanced model
    - increase model hyperparameters
    - Reduce amount of features – data could have too many features and the model is struggling to find patterns
    - Train Longer – sometimes they take longer

Types of Machine Learning Problems

- Fixes for overfitting
  - Collect more data
  - Try a less advanced model
    - The current model could model your data too well and remove the generalization
- When comparing models make sure the dataset is the same
- Keep the test sets separate at all cost do not use them in the validation and test data sets
- One best performance metric does not equal best model

Overfitting and Underfitting Definitions

- Poor performance on training data means the model hasn't learned properly
  - underfitting.
  - Try a different model, improve the existing one through hyperparameter or collect more data.
- Great performance on the training data but poor performance on test data means your model doesn't generalize well.
  - Your model may be overfitting the training data.
  - Try using a simpler model or making sure your the test data is of the same style your model is training on.
- Another form of overfitting can come in the form of better performance on test data than training data.
  - Reaons
    - This may mean your testing data is leaking into your training data (incorrect data splits)
    - you've spent too much time optimizing your model for the test set data.
  - Ensure your training and test datasets are kept separate at all times
  - avoid optimizing a models performance on the test set (use the training and validation sets for model improvement).

Tools We Will Use

- Anaconda – Data Science store for your computer
- Jupyter Notebooks
- Data Analysis
  - Pandas
  - Mathplot lib
  - Numpy
- Machine Learning
  - Tensor Flow
  - PyTorch
  - Scikit Learn
  - dmlc XGBoost
  - Cat Boost
- Jupyter Notebooks Jupyter and Anaconda