Masterclass on Tensorflow for beginners

By
Odemakinde Elisha Jesutofunmi

About Data Science Nigeria (DSN)

Data Science Nigeria is a non-profit driven by a vision to build a world-class Artificial Intelligence knowledge, research and innovation ecosystem that delivers high impact research, business use applications, locally-relevant AI-first start-ups, increase employability and drive for social good use cases.

Dr. Bayo Adekanmbi
CTO, MTN Nigeria,
Convener, Data Science Nigeria

For more about DSN kindly check us out at https://www.datasciencenigeria.org

About myself

Name: Odemakinde Elisha Jesutofunmi
Job: Data Scientist & Researcher
Where: Data Science Nigeria

Twitter: @elishatofunmi
Medium:@elishatofunmi
Github: @elishatofunmi
Email: elisha@datasciencenigeria.ai

CONTENT

- What's Next?
- Tensorflow prerequisites.
- Introduction to Tensorflow.
- Setting up tensorflow.
- Tensorflow operations (Basics).
- Neural Networks with Tensorflow (loss, optimizers, activation function, neurons).
- Putting it all together (titanic/house pricing).
- Model parameter hypertuning of neural nets.
- Tensorboard (visual).
- Useful links.

Source: here

What’s Next?

Source: here

What's Next?

Everyone tends to ask the question after trying their possible best on using all the machine learning algorithms they have ever known. Take note, not all machine learning problem requires deep learning but for some, it is mandatory.

What if trying a neural network will give you the best solution to your problem (Tensorflow is the first library you would love to experiment with in solving your problem).

Source: here

Tensorflow prerequisites

source: here

Tensorflow Prerequisites

- Must be convenient to write python efficiently.
- Good grasp of object oriented programming (OOP).
- Get familiar with numpy operations.
- Handling errors, debugging (your favorite as a scriptwriter).

Tensorflow Introduction

Source: here

Introduction to Tensorflow

Tensorflow is a symbolic math library and its used for machine learning applications such as neural networks, deep learning.

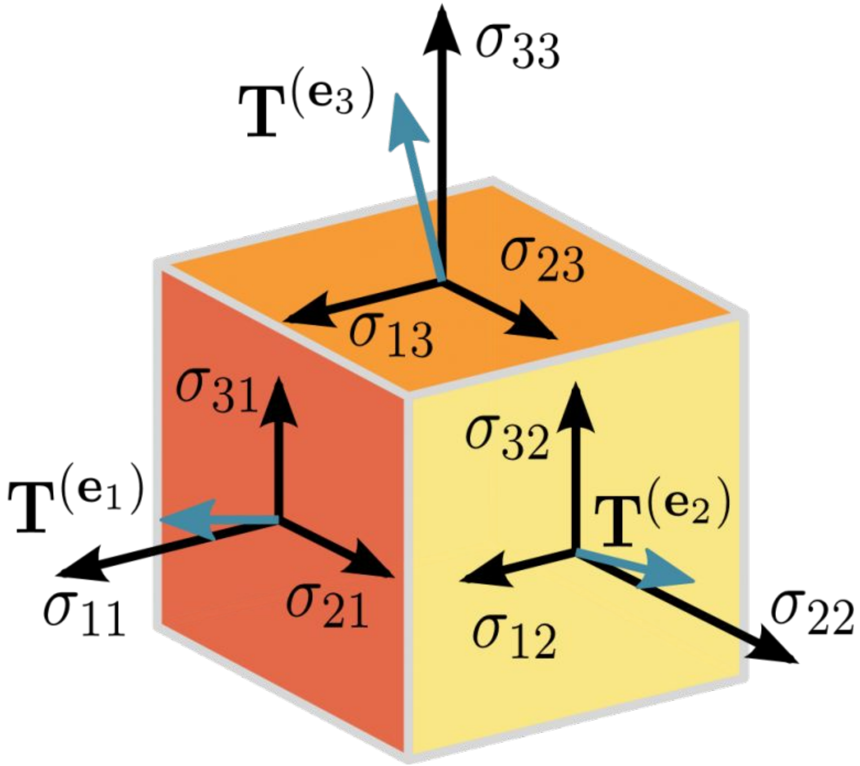
Source: here

Introduction to Tensorflow

Tensorflow was developed by the google brain team, the first version was released on the 11th of February, 2017. Its an open source library written in python, c++ and CUDA. It is a cross-platform application that is, it is available on Linux, macOS, Windows, android, JavaScript.

Introduction to Tensorflow

Tensors are containers that can house data in N dimensions. For example, matrices are 2 dimensional tensors.
We can further say, a tensor is a generalization of matrices to N-dimensional space.
This implies, with tensors we can visualize N-dimensional space vectors.

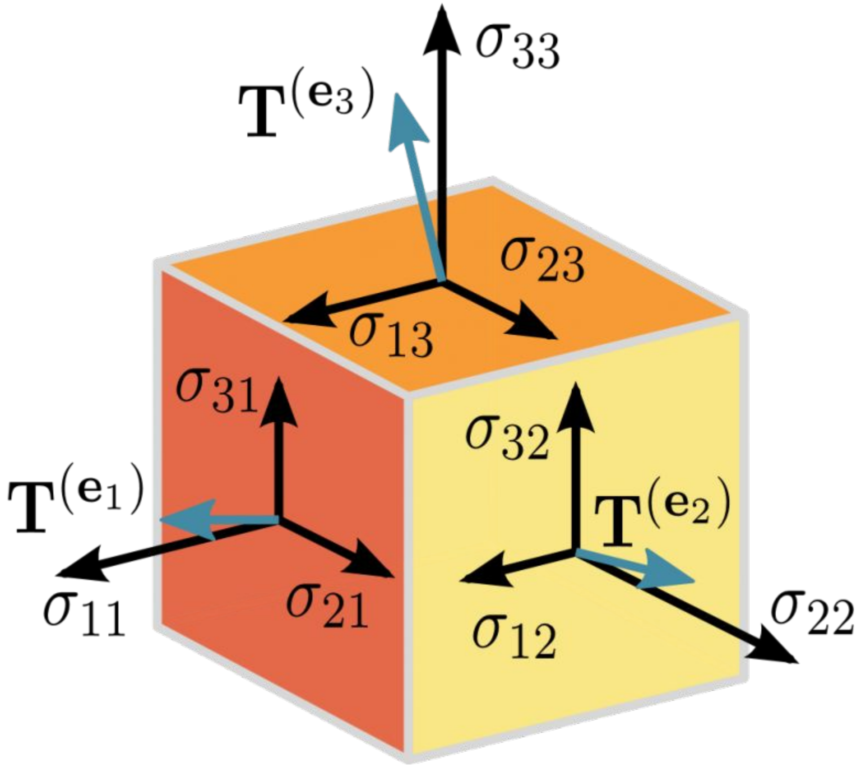


Source: [here]

Introduction to Tensorflow

How tensors differ from numpy (multi-dimensional array).

- Tensors are containers or function that needs to be defined.
- Numpy can be referred to as generic tensor.



Setting up Tensorflow

Setting up Tensorflow

- Setting up on windows/linux via conda (virtual environment).
- pip install tensorflow==2.0 (conda/ command prompt).

Tensorflow Operations

Tensorflow Operations

Importing Tensorflow:

```
import tensorflow as tf
```

Tensorflow Operations

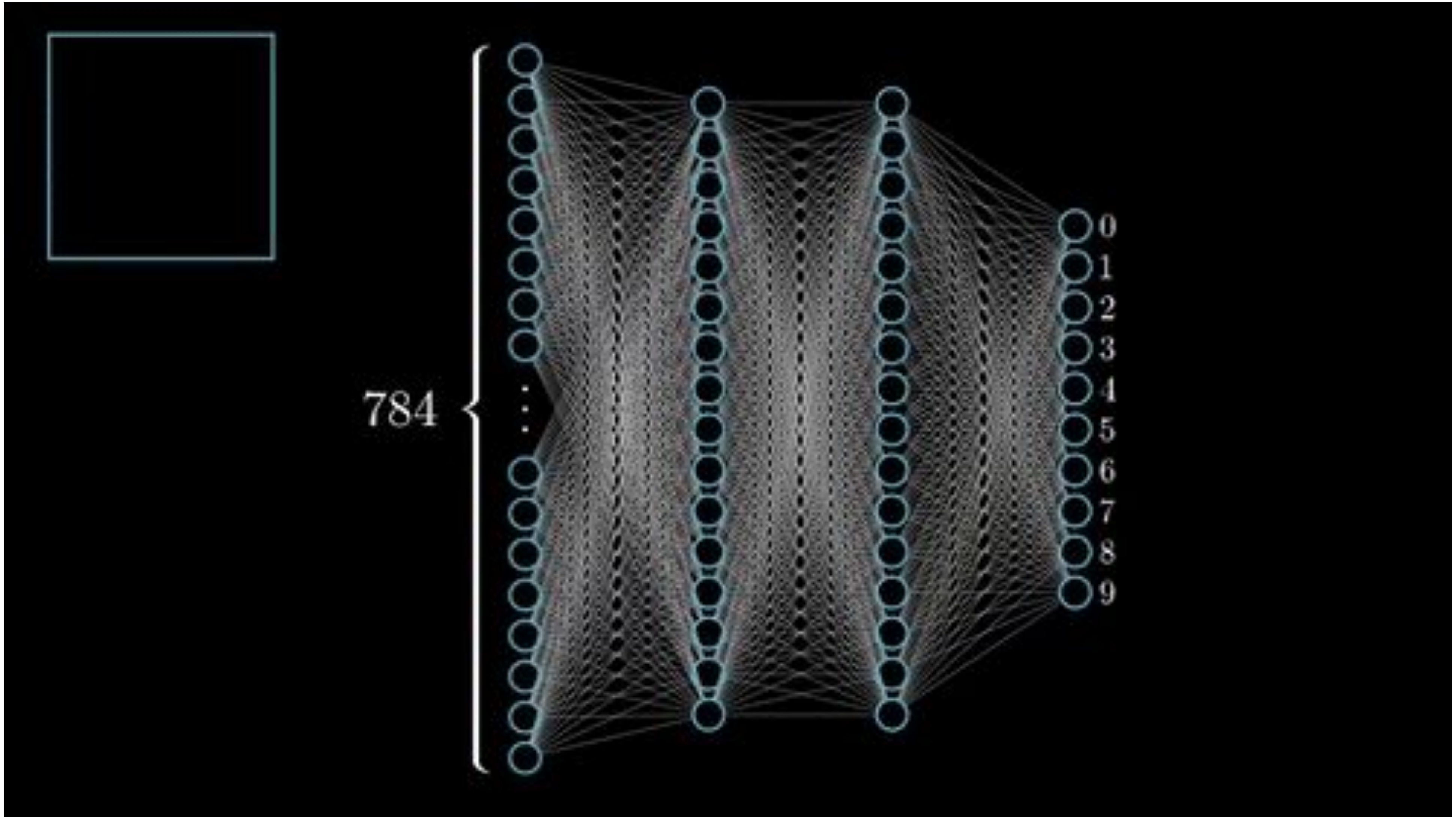
- Data types.
- tf.tensors.
- tf.variable.
- tf.placeholder.
- tf.add
- tf.matmul
- tf.zeros, tf.ones
- tf.confusion_matrix

TensorFlow

Neural Networks with Tensorflow (loss, optimizers, activation function, neurons).

TensorFlow

Neural Networks (neurons and layers)



source: here

Neural Network (Activation Functions)

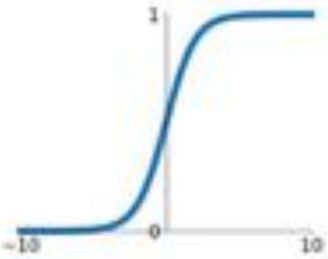
Activation functions are computational functions for the neuron cells. They are algorithms that enhances learning in each neuron cell. They include:
Relu, softmax, sigmoid, elu, maxout, tanh etc.

Neural Network (Activation Functions)

Activation Functions

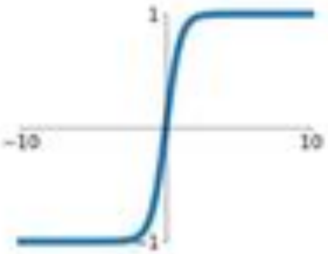
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



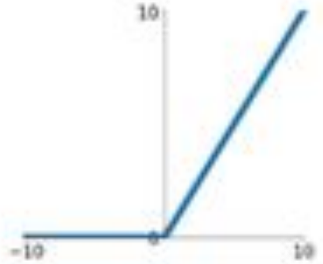
tanh

$\tanh(x)$



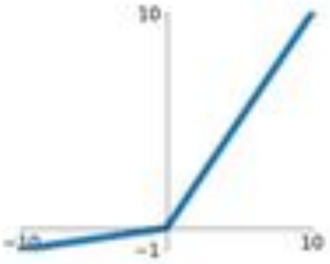
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

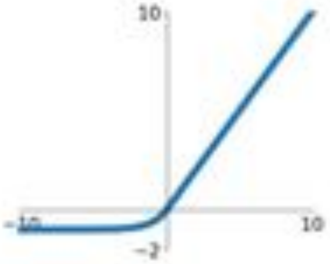


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



source: [here]

Neural Networks (losses)

This includes:

1. Binary cross entropy (computes the loss between true labels and predicted values).
2. Categorical cross entropy (computes the loss between labels and predicted classes).

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Source: here

$$CCE = -\frac{1}{N}\sum_{i=0}^{N}\sum_{j=0}^{J} y_j \cdot log(\hat{y}_j) + (1 - y_j) \cdot log(1 - \hat{y}_j)$$

Source: here

Neural Networks (losses)

This includes:

1. Cosine similarity (Computes the categorical hinge loss between y_true and y_pred).
2. mean absolute error (Computes the mean of absolute difference between labels and predictions).

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Source: here

HIOX

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

©easycalculation.com

Source: here

Neural Networks (losses)

This includes:

1. mean absolute percentage error(Computes the mean absolute percentage error between y_true and y_pred).
2. mean squared error (Computes the mean of squares of errors between labels and predictions).

$$MAPE = \frac{100\%}{N} \sum_{i=0}^{N} \frac{|y_i - \hat{y}_i|}{\hat{y}_i}$$

Source: here

Source: here

Neural Networks (Optimizers)

Optimizers are algorithms. Examples includes:

1. adadelta.
2. Adam.
3. RMSprop.
4. Adamax.
5. Adagrad.
6. Nadem.
7. SGD.

$$\mathbf{m} \leftarrow \beta_1 \mathbf{m} + (1 - \beta_1)\nabla_\theta J(\theta)$$

$$\mathbf{s} \leftarrow \beta_2 \mathbf{s} + (1 - \beta_2)\nabla_\theta J(\theta) \otimes \nabla_\theta J(\theta)$$

$$\mathbf{m} \leftarrow \frac{\mathbf{m}}{1 - {\beta_1}^T}$$

$$\mathbf{s} \leftarrow \frac{\mathbf{s}}{1 - {\beta_2}^T}$$

$$\theta \leftarrow \theta - \eta\, \mathbf{m} \oslash \sqrt{\mathbf{s} + \epsilon}$$

Adam optimizer (algorithm)

Neural Networks (Optimizers)

Optimizers are algorithms. Examples includes:

1. adadelta.
2. Adam.
3. RMSprop.
4. Adamax.
5. Adagrad.
6. Nadem.
7. SGD.

$$\mathbf{s} \leftarrow \beta \mathbf{s} + (1 - \beta) \nabla_\theta J(\theta) \otimes \nabla_\theta J(\theta)$$

$$\theta \leftarrow \theta - \eta \nabla_\theta J(\theta) \oslash \sqrt{\mathbf{s} + \epsilon}$$

RMSprop optimizer (algorithm)

$$\mathbf{s} \leftarrow \mathbf{s} + \nabla_\theta J(\theta) \otimes \nabla_\theta J(\theta)$$

$$\theta \leftarrow \theta - \eta \nabla_\theta J(\theta) \oslash \sqrt{\mathbf{s} + \epsilon}$$

Adagrad optimizer (algorithm)

Neural Networks (Optimizers)

Optimizers are algorithms. Examples includes:

1. adadelta.
2. Adam.
3. RMSprop.
4. Adamax.
5. Adagrad.
6. Nadem.
7. SGD.

Gradient descent algorithm

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\
&\qquad (\text{for } j = 1 \text{ and } j = 0) \\
&\}
\end{aligned}
$$

$$
w_k \rightarrow w'_k = w_k - \eta \frac{\partial C}{\partial w_k}
$$

$$
b_l \rightarrow b'_l = b_l - \eta \frac{\partial C}{\partial b_l}.
$$

Gradient descent algorithm

putting it all together
(titanic/ house pricing)

TensorFlow

Model parameter hypertuning (neural networks)

Tensorboards

TensorBoard

Tensorboard gives us the flexibility of visualizing the training parameters, metrics or any statistics in our Neural Network.

TensorBoard is a tool for providing the measurements and visualizations needed during the machine learning workflow. It enables tracking experiment metrics like loss and accuracy, visualizing the model graph, projecting embeddings to a lower dimensional space, and much more.(https://www.tensorflow.org).

TensorBoard

With tensorboard you can visualize the following:

1. Scalers (something like accuracy of your predictions).
2. Neural network graph (Visualize your neural network model).
3. Distribution (visualize how data changes, that is, weights and biases).
4. Input Data (images, audio, text etc).

Reference Materials

- https://docs.anaconda.com/anaconda/user-guide/tasks/tensorflow/
- https://www.easy-tensorflow.com/tf-tutorials/basics/graph-and-session
- https://www.tensorflow.org
- https://www.analyticsvidhya.com/blog/2017/03/tensorflow-understanding-tensors-and-graphs/
- https://machinelearningmastery.com/introduction-to-tensors-for-machine-learning/
- https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/
- https://www.kaggle.com/c/caterpillar-tube-pricing/data
- https://www.datacamp.com/community/tutorials/tensorboard-tutorial

Reference Materials

- https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/
- https://www.tensorflow.org/learn
- https://www.tensorflow.org/guide/data
- https://www.tensorflow.org/guide/keras/overview
- https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f
- http://miguelromao.me/2018/03/19/a-simple-low-level-tensorflow-classifier/
- https://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/
- https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

Website: www.datasciencenigeria.org
Email: info@datasciencenigeria.org
Twitter: Datasciencenig
Instagram: Datasciencenigeria
Facebook: facebook.com/datasciencenig
YouTube: https://goo.gl/Vcjjyp
Phone : +2348140000853, +2348140033300

For videos and other reports, you can see a link to 25 key milestones of the non-profit via this link https://goo.gl/Hc5Bhd