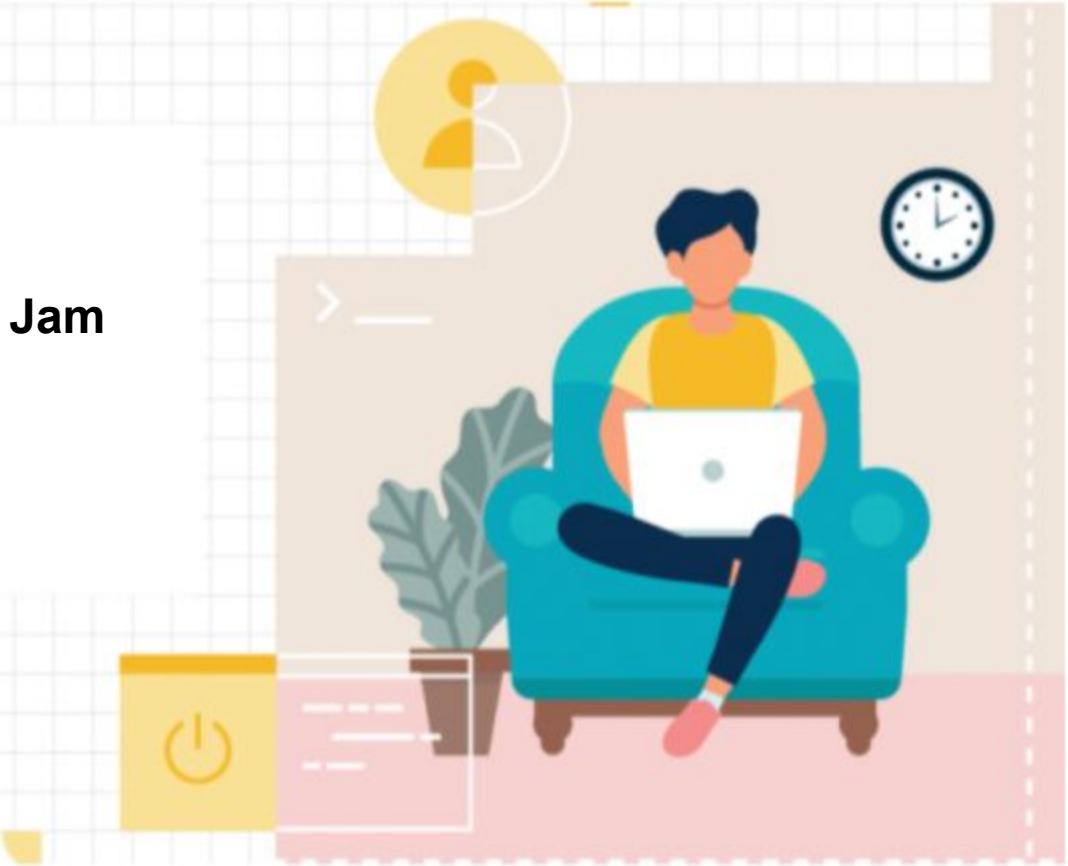
Machine Learning Study Jam

Introduction to Deep Learning



Outline

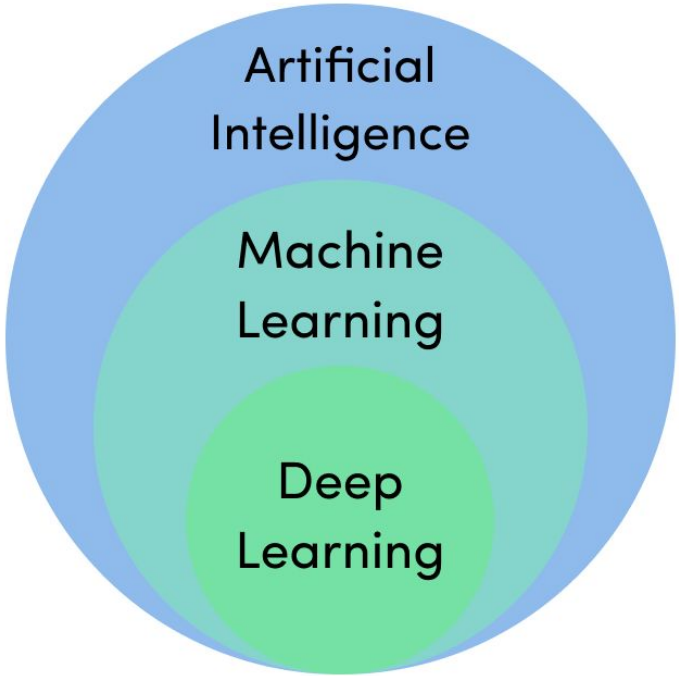
1. Introduction to Deep Learning and Computer Vision

2. Practice Exercises:
   a. Learn TensorFlow 1: Say hello to the "Hello, World" of machine learning
   b. Learn Tensorflow 2: Build a computer vision model with TensorFlow

3. Wrap Up

4. What’s Next

Introduction to Deep Learning



Artificial Intelligence (AI)

- Incorporating human intelligence to machines.
- Human can see things => computer vision
- Human can speak => natural language processing

Machine Learning (ML)

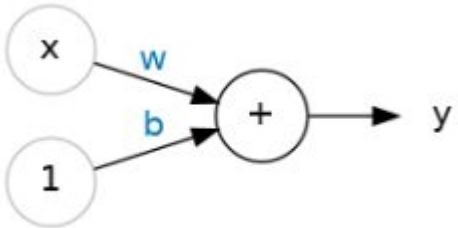
- Subset of AI, empowering computer systems with the ability to “learn”.
- Supervised learning, unsupervised learning, reinforcement learning

Deep Learning (DL)

- Subset of ML, inspired by brain’s structure and function called artificial neural network.

The Linear Unit

- Neuron - The fundamental component of a neural network.



The Linear Unit: $y = wx + b$

- The input is x. Its connection to the neuron has a **weight** which is w. Whenever a value flows through a connection, you multiply the value by the connection's weight. For the input x, what reaches the neuron is `w * x`. A neural network "learns" by modifying its weights.
- The b is a special kind of weight we call the **bias**. The bias doesn't have any input data associated with it; instead, we put a 1 in the diagram so that the value that reaches the neuron is just b (since `1 * b = b`). The bias enables the neuron to modify the output independently of its inputs.
- The y is the value the neuron ultimately outputs. To get the output, the neuron sums up all the values it receives through its connections. This neuron's activation is `y = w * x + b`, or as a formula $y=wx+b$

Introduction

Introduction - What is Machine Learning?

Consider the following problem: You're building a system that performs activity recognition for fitness tracking.




Introduction - What is Machine Learning?

Consider the following problem: You're building a system that performs activity recognition for fitness tracking.

```
if(speed<4){
  status=WALKING;
}
```

Introduction - What is Machine Learning?

Consider the following problem: You're building a system that performs activity recognition for fitness tracking.

```
if(speed<4){
  status=WALKING;
}
```

```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

Introduction - What is Machine Learning?

Consider the following problem: You're building a system that performs activity recognition for fitness tracking.

```
if(speed<4){
  status=WALKING;
}
```

```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

```
if(speed<4){
    status=WALKING;
} else if(speed<12){
    status=RUNNING;
} else {
    status=BIKING;
}
```

Introduction - What is Machine Learning?

Consider the following problem: You're building a system that performs activity recognition for fitness tracking.

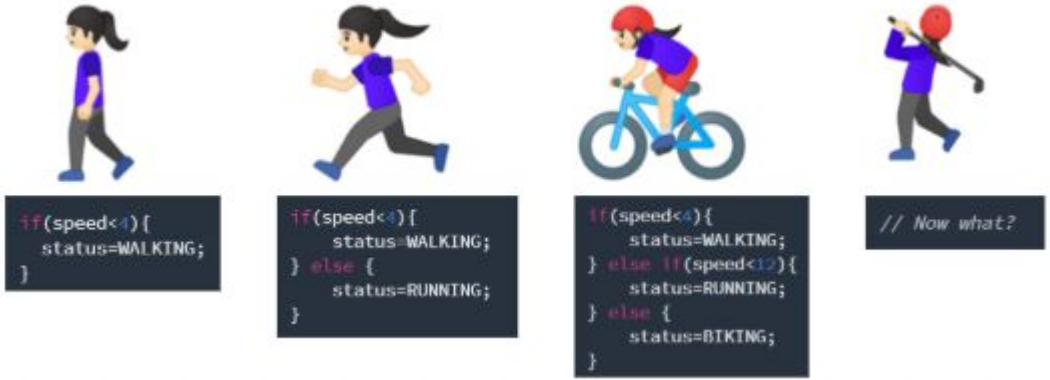
```
if(speed<4){
  status=WALKING;
}
```

```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

```
if(speed<4){
    status=WALKING;
} else if(speed<12){
    status=RUNNING;
} else {
    status=BIKING;
}
```

```
// Now what?
```

Introduction - What is Machine Learning?



Introduction - What is Machine Learning?

Consider the following problem: You're building a system that performs activity recognition for fitness tracking.

```
01010010101001010101
10010101010010111101
01001010100101010010
01010010101001010100
```

Label = WALKING

```
10101001010010101010
01010100100100100001
00100111111010101111
10101001001111010111
```

Label = RUNNING

```
10010100111110101010
11010101110101011100
10101011110101010111
11111100011110101010
```

Label = BIKING

```
11111111110100111010
00111111010111110101
01011101010101011100
10101010101001111100
```

Label = GOLFING
(Sort of)

Introduction - What is Machine Learning?

Introduction - What is Machine Learning?

Consider the result of this to be a model, which at runtime is used like this:

Learn TensorFlow 1:

The "Hello World" of machine learning

Learn TensorFlow 1: The "Hello World" of machine learning

Consider the following sets of numbers. Can you see the relationship between them?

X:	-1	0	1	2	3	4
Y:	-2	1	4	7	10	13

Learn TensorFlow 1: The "Hello World" of machine learning

Consider the following sets of numbers. Can you see the relationship between them?

X:	-1	0	1	2	3	4
Y:	-2	1	4	7	10	13

Answer: Y=3X+1

Learn TensorFlow 1: The "Hello World" of machine learning

Consider the following sets of numbers. Can you see the relationship between them?

X:	-1	0	1	2	3	4
Y:	-2	1	4	7	10	13

Answer: Y=3X+1

```
float my_function(float x){
    float y = (3 * x) + 1;
    return y;
}
```

Learn TensorFlow 1:

The "Hello World" of machine learning

(Practice Exercise)

Create new notebook:

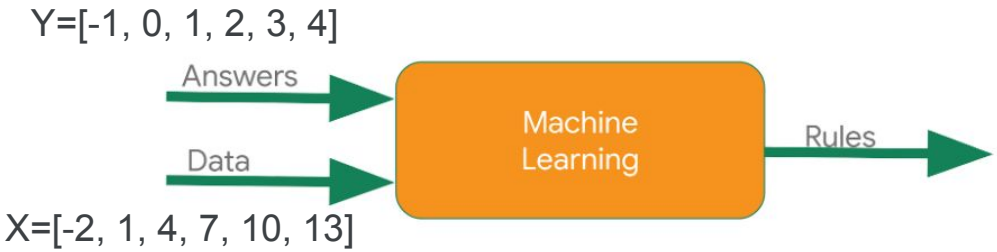
https://colab.research.google.com/

URL: https://tinyurl.com/MLSJ-lab1

Learn TensorFlow 1: The "Hello World" of machine learning (Practice Exercise)

X:	-1	0	1	2	3	4
Y:	-2	1	4	7	10	13

Training



Consider the result of this to be a model, which at runtime is used like this:

Testing

Prerequisites

Before attempting this codelab, you'll want to have:

- A solid knowledge of Python
- Basic programming skills

What you'll need

If you've never created an ML model using TensorFlow, you can use Colaboratory, a browser-based environment containing all the required dependencies.

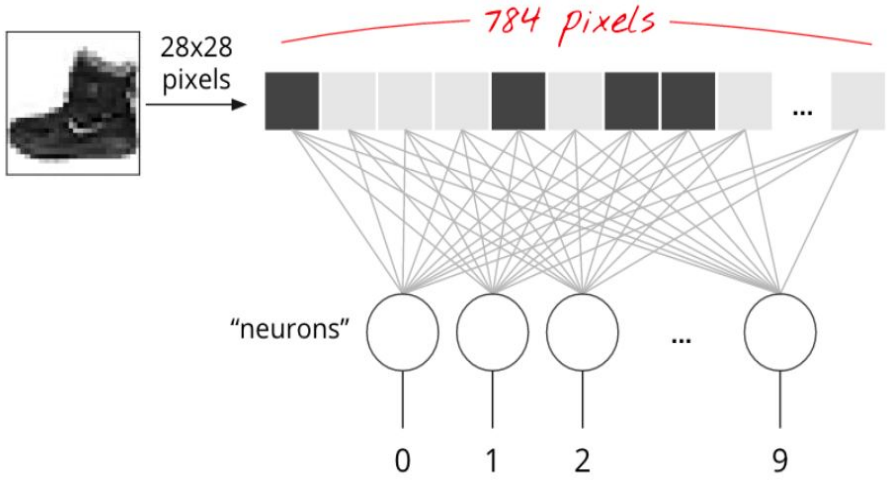
Learn TensorFlow 2:

Build a computer vision model with TensorFlow (Practice Exercise)

Neural Network 101

Fashion MNIST dataset are 28x28 pixel grayscale images. The simplest approach for classifying them is to use the 28x28=784 pixels as inputs for a 1-layer neural network.

Each "**neuron**" in a neural network does a weighted sum of all of its inputs, adds a constant called the "**bias**" and then feeds the result through some non-linear "**activation function**". The "**weights**" and "**biases**" are parameters that will be determined through training. They are initialized with random values at first.



The picture above represents a 1-layer neural network with 10 output neurons since we want to classify fashion MNIST images into 10 classes (0 to 9).

Neural Network 101

Glossary

batch or **mini-batch**: training is always performed on batches of training data and labels. Doing so helps the algorithm converge. The "batch" dimension is typically the first dimension of data tensors. For example a tensor of shape [100, 192, 192, 3] contains 100 images of 192x192 pixels with three values per pixel (RGB).

cross-entropy loss: a special loss function often used in classifiers.

dense layer: a layer of neurons where each neuron is connected to all the neurons in the previous layer.

features: the inputs of a neural network are sometimes called "features". The art of figuring out which parts of a dataset (or combinations of parts) to feed into a neural network to get good predictions is called "feature engineering".

labels: another name for "classes" or correct answers in a supervised classification problem

learning rate: fraction of the gradient by which weights and biases are updated at each iteration of the training loop.

logits: the outputs of a layer of neurons before the activation function is applied are called "logits". The term comes from the "logistic function" a.k.a. the "sigmoid function" which used to be the most popular activation function. "Neuron outputs before logistic function" was shortened to "logits".

Neural Network 101

Glossary

loss: the error function comparing neural network outputs to the correct answers

neuron: computes the weighted sum of its inputs, adds a bias and feeds the result through an activation function.

one-hot encoding: class 3 out of 5 is encoded as a vector of 5 elements, all zeros except the 3rd one which is 1.

relu: rectified linear unit. A popular activation function for neurons.

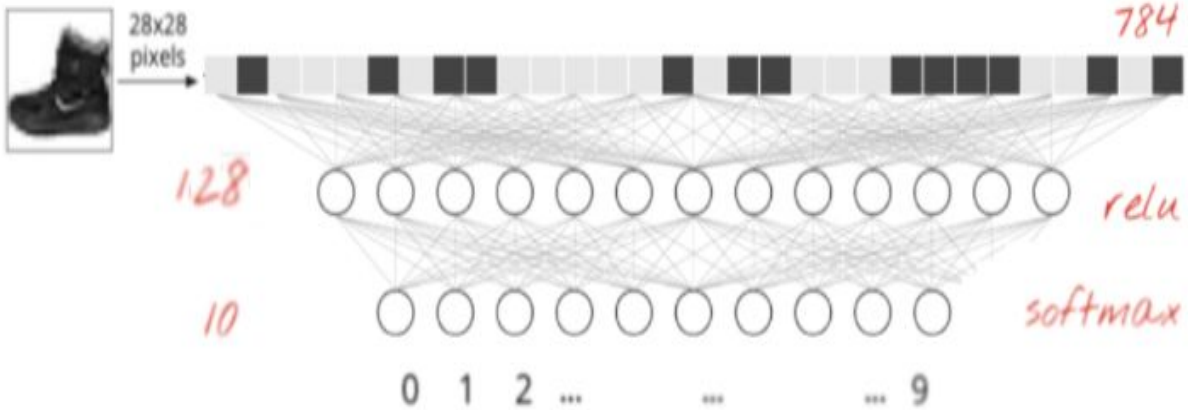
sigmoid: another activation function that used to be popular and is still useful in special cases.

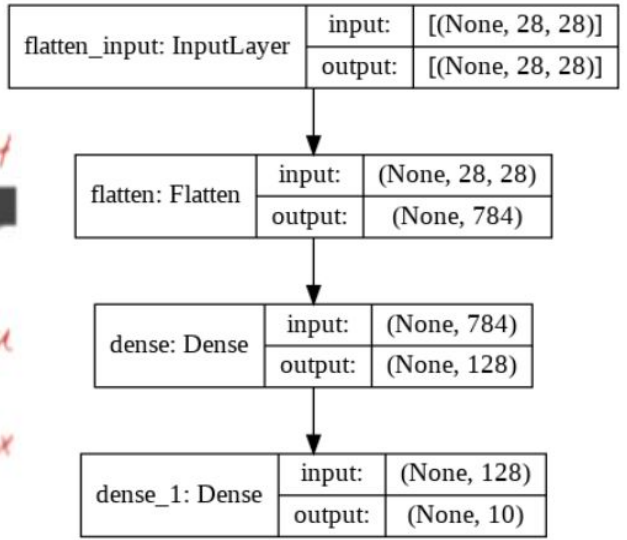
softmax: a special activation function that acts on a vector, increases the difference between the largest component and all others, and also normalizes the vector to have a sum of 1 so that it can be interpreted as a vector of probabilities. Used as the last step in classifiers.

tensor: A "tensor" is like a matrix but with an arbitrary number of dimensions. A 1-dimensional tensor is a vector. A 2-dimensions tensor is a matrix. And then you can have tensors with 3, 4, 5 or more dimensions.

Neural Network 101

```
model = tf.keras.models.Sequential([tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(128, activation=tf.nn.relu),
                                    tf.keras.layers.Dense(10, activation=tf.nn.softmax)])
```

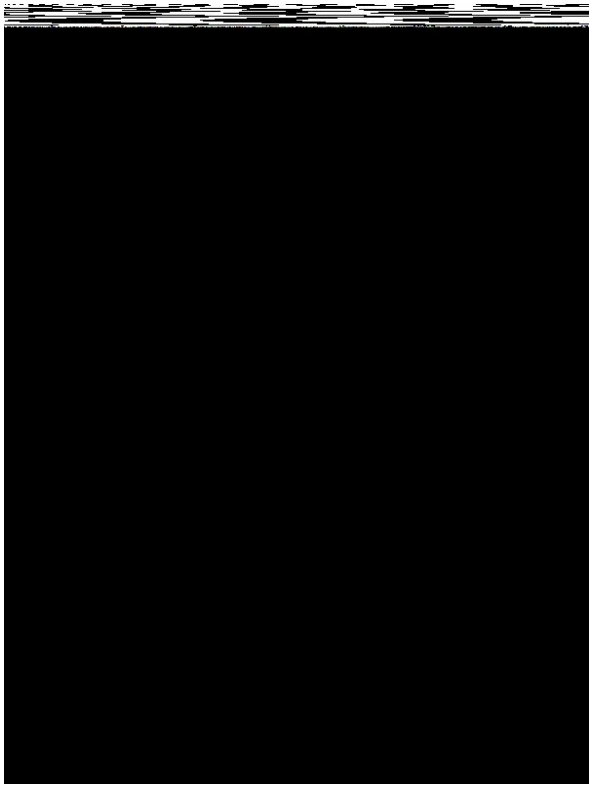




The first layer in this network, **tf.keras.layers.Flatten**, transforms the format of the images from a two-dimensional array (of 28 by 28 pixels) to a one-dimensional array (of 28 * 28 = 784 pixels). Think of this layer as unstacking rows of pixels in the image and lining them up. This layer has no parameters to learn; it only reformats the data.

After the pixels are flattened, the network consists of a sequence of two **tf.keras.layers.Dense** layers. These are densely connected, or fully connected, neural layers. The first Dense layer has 128 nodes (or neurons). The second (and last) layer returns a logits array with length of 10. Each node contains a score that indicates the current image belongs to one of the 10 classes.

Neural Network 101



INPUT IMAGE

18	54	51	239	244	188
55	121	75	78	95	88
35	24	204	113	109	221
3	154	104	235	25	130
15	253	225	159	78	233
68	85	180	214	245	0

WEIGHT

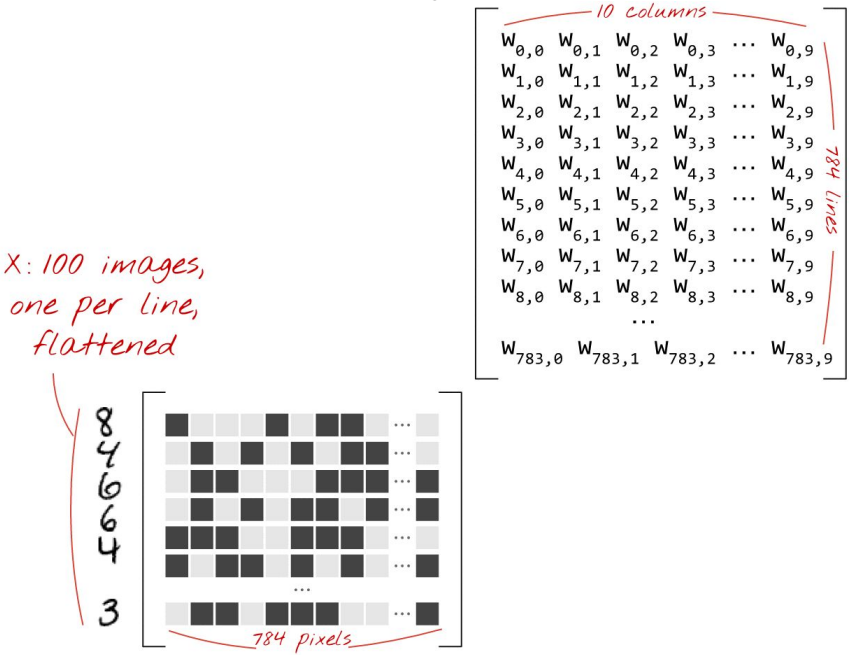
1	0	1
0	1	0
1	0	1

429

https://medium.com/@adityaraj_64455/it-all-started-with-cnns-alexnet-3023b21bb891

Neural Network 101

Here is how a neural network layer, processing a collection of images, can be represented by a matrix multiplication:

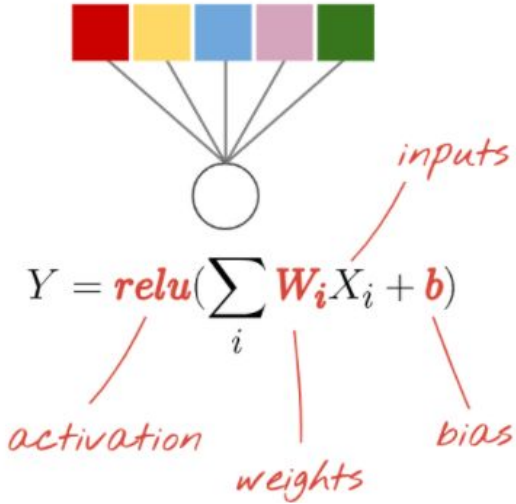


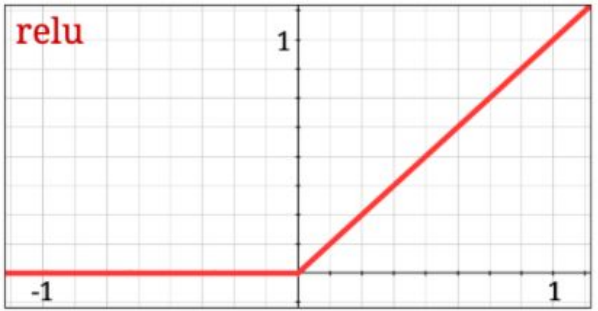
Using the first column of weights in the weights matrix W, we compute the weighted sum of all the pixels of the first image. This sum corresponds to the first neuron. Using the second column of weights, we do the same for the second neuron and so on until the 10th neuron. We can then repeat the operation for the remaining 99 images. If we call X the matrix containing our 100 images, all the weighted sums for our 10 neurons, computed on 100 images are simply X.W, a matrix multiplication.

Neural Network 101

Activation functions: relu

You would typically use the "relu" activation function for all layers but the last.





Again, a "**neuron**" computes a weighted sum of all of its inputs, adds a value called "**bias**" and feeds the result through the activation function.

The most popular activation function is called "**RELU**" for Rectified Linear Unit. It is a very simple function as you can see on the graph above.

Neural Network 101

Softmax activation for classification

The last layer of our neural network has 10 neurons because we want to classify handwritten digits into 10 classes (0,..9). It should output 10 numbers between 0 and 1 representing the probability of this digit being a 0, a 1, a 2 and so on. For this, on the last layer, we will use an activation function called "**softmax**".

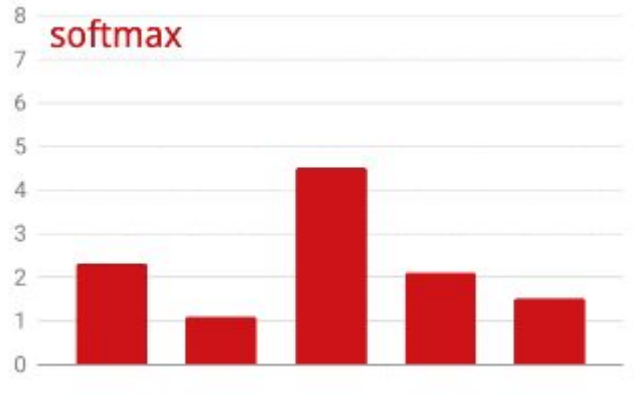
Applying softmax on a vector is done by taking the exponential of each element and then normalising the vector, typically by dividing it by its "**L1**" norm (i.e. sum of absolute values) so that normalized values add up to 1 and can be interpreted as probabilities.

The output of the last layer, before activation is sometimes called "**logits**". If this vector is L = [L0, L1, L2, L3, L4, L5, L6, L7, L8, L9], then:

$$softmax(L_n) = \frac{e^{L_n}}{||e^{L}||}$$

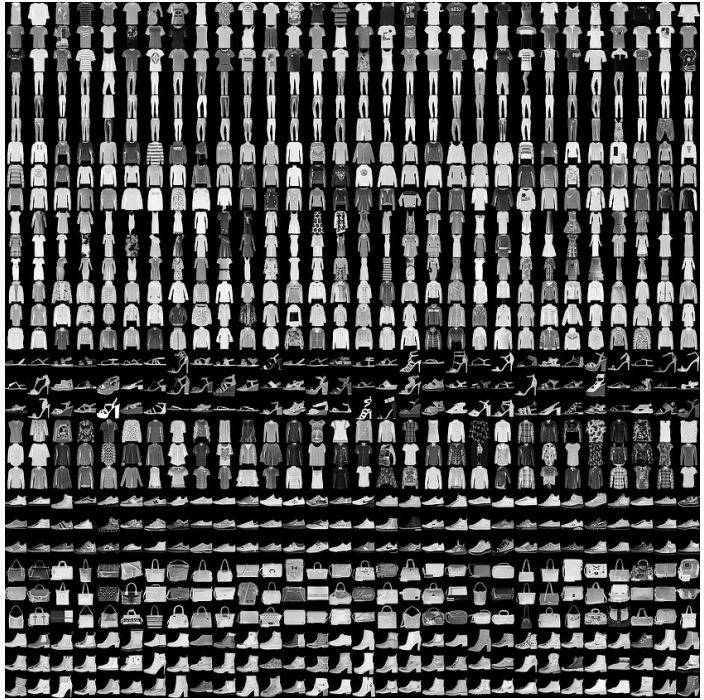
weighted sum+bias

L1 norm



Learn TensorFlow 2: Build a computer vision model with TensorFlow (Practice Exercise)

https://knowyourdata-tfds.withgoogle.com/#tab=STATS&dataset=fashion_mnist



Wrap Up

TensorFlow Study Jam - Summary

1. Learn TensorFlow 1: Say hello to the "Hello, World" of machine learning

   What you've learnt

   - The basics of machine learning

   What you've built

   - Your first machine learning model

2. Learn Tensorflow 2: Build a computer vision model with TensorFlow

   What you've learnt

   - Train a neural network to recognize articles of clothing
   - Complete a series of exercises to guide you through experimenting with the different layers of the network

   What you've built

   - A neural network that identifies articles of clothing

TensorFlow Study Jam - Learning Materials

1. Learn TensorFlow 1: Say hello to the "Hello, World" of machine learning

   https://developers.google.com/codelabs/tensorflow-1-helloworld#0

2. Learn Tensorflow 2: Build a computer vision model with TensorFlow

   https://developers.google.com/codelabs/tensorflow-2-computervision#0

Q&A

What’s Next

TensorFlow Study Jam - Learning Materials (More...)

1. Learn Tensorflow 3: Build convolutions and perform pooling

   https://developers.google.com/codelabs/tensorflow-3-convolutions#0

2. Learn Tensorflow 4: Build convolutional neural networks (CNNs) to enhance computer vision

   https://developers.google.com/codelabs/tensorflow-4-cnns#0

3. Learn Tensorflow 5: Use convolutional neural networks (CNNs) with complex images

   https://developers.google.com/codelabs/tensorflow-5-compleximages#0

4. Learn Tensorflow 6: Use convolutional neural networks (CNNs) with large datasets to avoid overfitting

   https://developers.google.com/codelabs/tensorflow-6-largecnns#0

TensorFlow Study Jam - Learning Materials (More...)

1. TensorFlow, Keras and deep learning, without a PhD

   https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist

2. Google Machine Learning Crash Course

   https://developers.google.com/machine-learning/crash-course

3. What is Neural Network

   https://www.youtube.com/watch?v=aircAruvnKk

4. Google Neural Network Playground

   https://playground.tensorflow.org/

Thank You!