

neural
networks

neuron

- calculus
- probability
- STATISTICS

stochastic
gradient
descent

keras

- sequential
- layers
- loss function
- optimizer

$$y = 2x - 1$$

$$x = -1, 0, 1, 2, 3, 4$$

$$y = -3, -1, 1, 3, 5, 7$$

$$y = 3x + 0$$

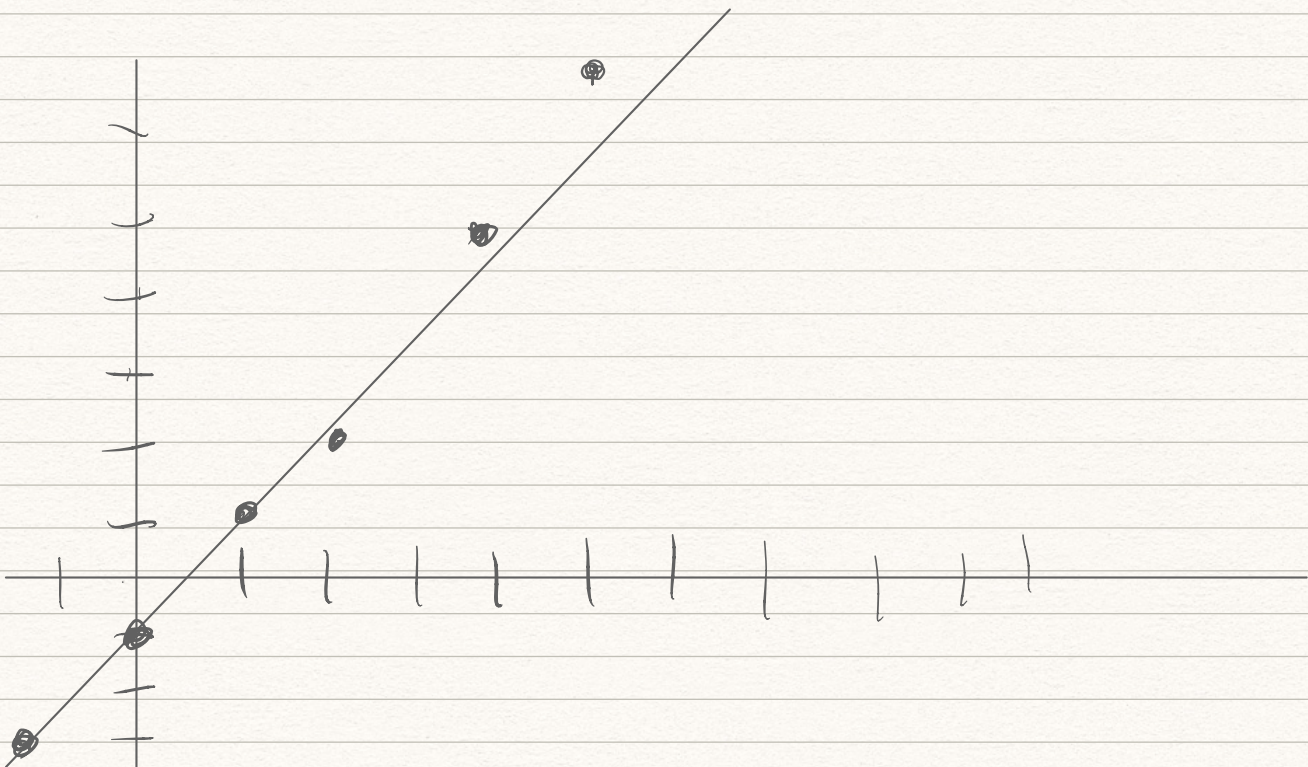
$$x = -1 \Rightarrow y = 3(-1) + 0 = -3$$



$$x = 0 \Rightarrow y = 3(0) + 0 = 0$$



$$x = 1 \Rightarrow y = 3(1) + 0 = 3$$



positive

steeper than $\frac{1}{1}$, $(1.5, z=m)$

```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])  
model.compile(optimizer='sgd', loss='mean_squared_error')
```

then 1 dense \Rightarrow 1 layer \Rightarrow single "neuron"

model.compile

(optimizer = 'sgd', loss = mean_squared_error)

```
model.compile(optimizer='sgd', loss='mean_squared_error')
```

guess \rightarrow measure
how good
how bad

Stochastic
gradient
descent

python numpy

```
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
```

```
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)
```

```
model.fit(xs, ys, epochs=500)
```

```
print(model.predict([10.0]))
```

the predicted ^{or} value of (10) with
insert data