#ODSC

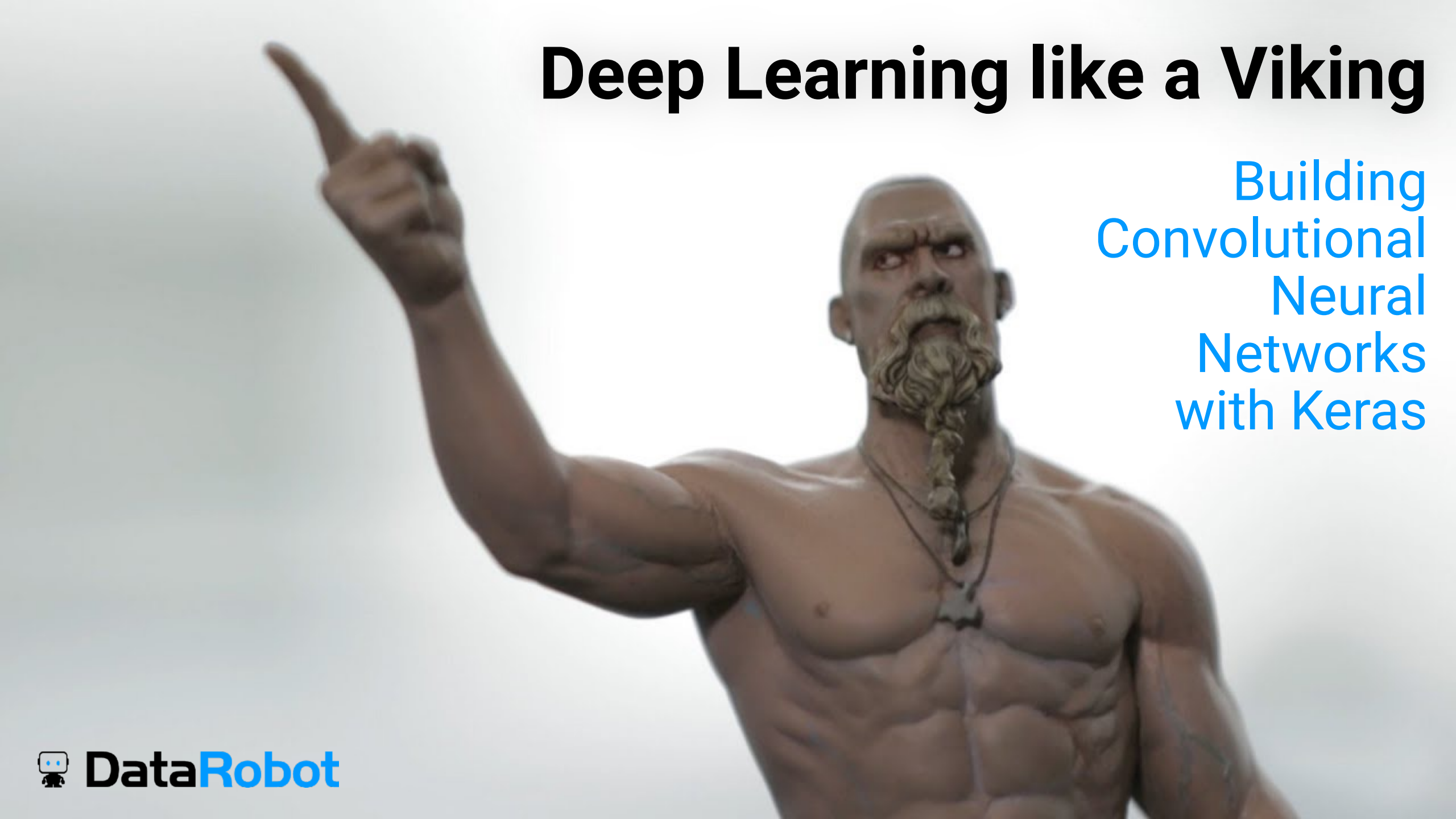
BOSTON
APR 30 – MAY 3

Deep Learning like a Viking: Building Convolutional Neural Networks with Keras

Guy Royse
Developer Evangelist,
DataRobot

Deep Learning like a Viking

Building Convolutional Neural Networks with Keras



DataRobot

Guy Royse

Developer Evangelist

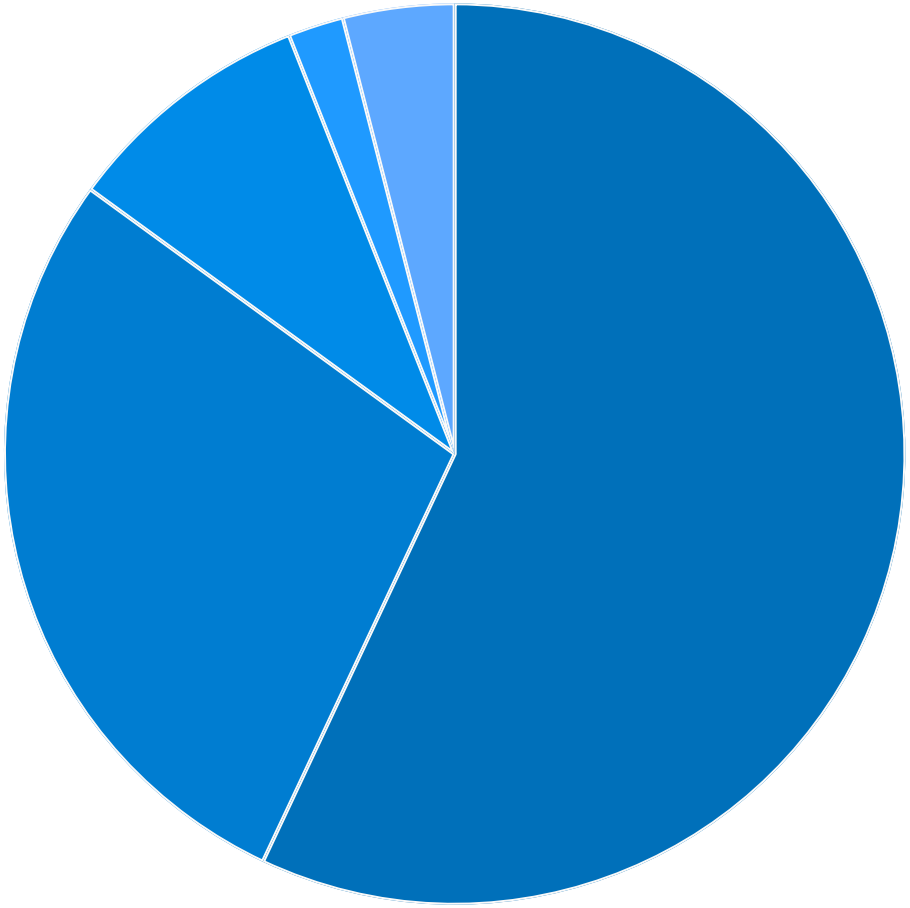
DataRobot

guyroyse

code.guy.dev

guy.dev

IANADS



British Isles	57%
German	28%
Iberian	9%
Uncertain	4%
Scandinavian	2%
Guy	100%

Lingsberg Runestones

Danr and Húskarl and Sveinn and Holmfríðr, the mother and (her) sons, had this stone erected in memory of Halfdan, the father of Danr and his brothers; and Holmfríðr in memory of her husbandman.

The Younger Futhark

ᚠ	ᚢ	ᚦ	ᚬ	ᚱ	ᚴ	ᚼ	ᚾ	ᛁ	ᛅ	ᛋ	ᛏ	ᛒ	ᛘ	ᛚ	ᛦ
fe	ur	thurs	as	reith	kaun	hagall	nauthr	isa	ar	sol	tyr	bjork	mathr	logr	yr

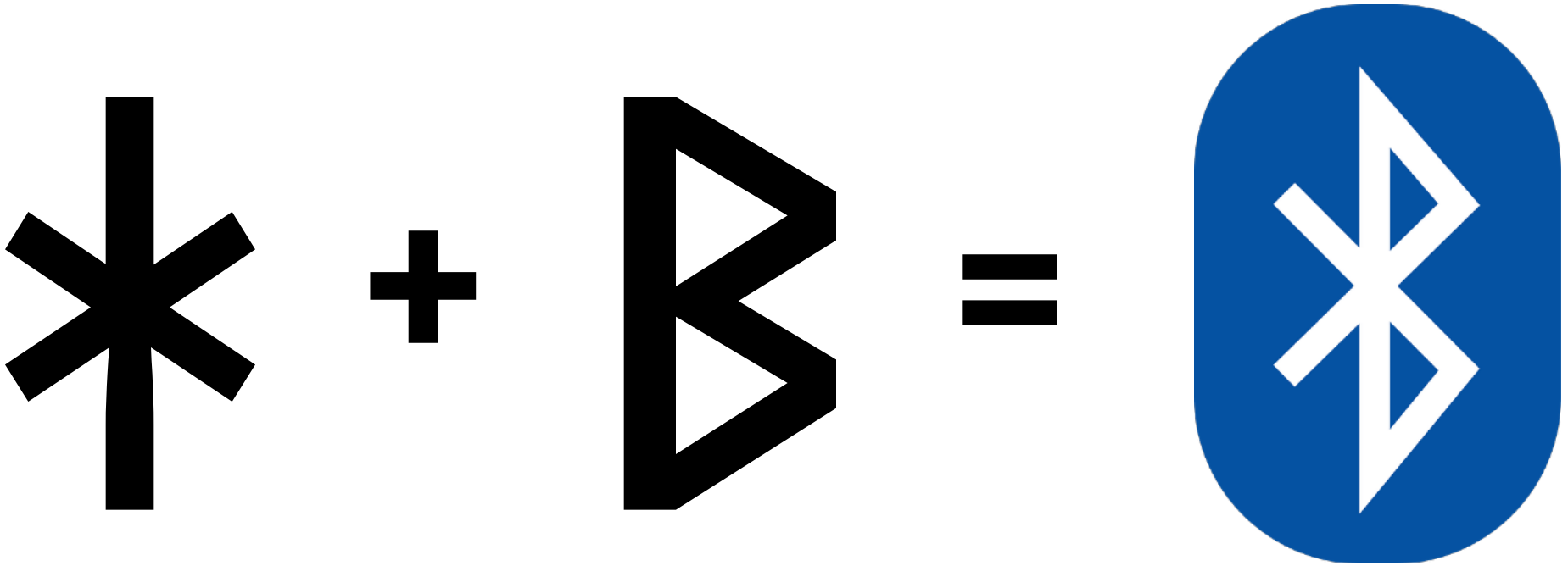
ᚴᚬᛁᛋᚱᚬᛁᛋ

(Kai Rais)

Lingsberg Runestones

Danr and Húskarl and Sveinn and **Holmfríðr**, the mother and (her) sons, had this stone erected in memory of Halfdan, the father of Danr and his brothers; and **Holmfríðr** in memory of her husbandman.

Harald Bluetooth



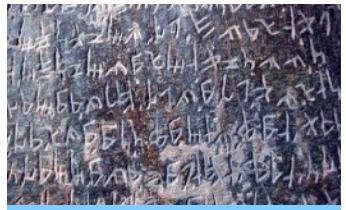
Younger Futhark vs. Latin & Greek

ᚠ	ᚢ	ᚦ	ᚭ	ᚱ	ᚴ	ᚼ	ᚾ	ᛁ	ᛅ	ᛋ	ᛏ	ᛒ	ᛘ	ᛚ	ᛦ
fe	ur	thurs	as	reith	kaun	hagall	nauthr	isa	ar	sol	tyr	bjork	mathr	logr	yr

F	U			R				I		S	T	B		L	
		Θ		P				I		Σ	T	B		Λ	

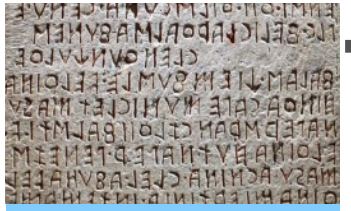
Common Ancestors

Egyptian

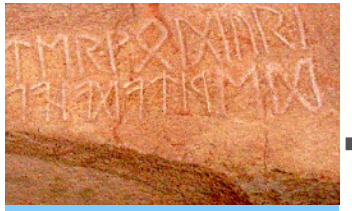


Proto-Canaanite

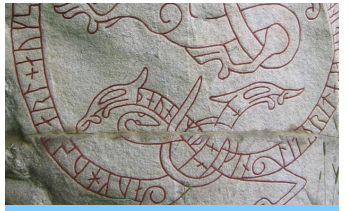
Ancient Greek



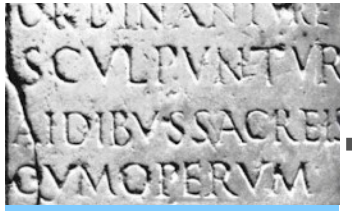
Etruscan



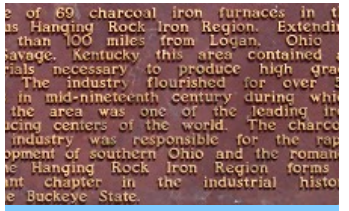
Elder Futhark



Younger Futhark

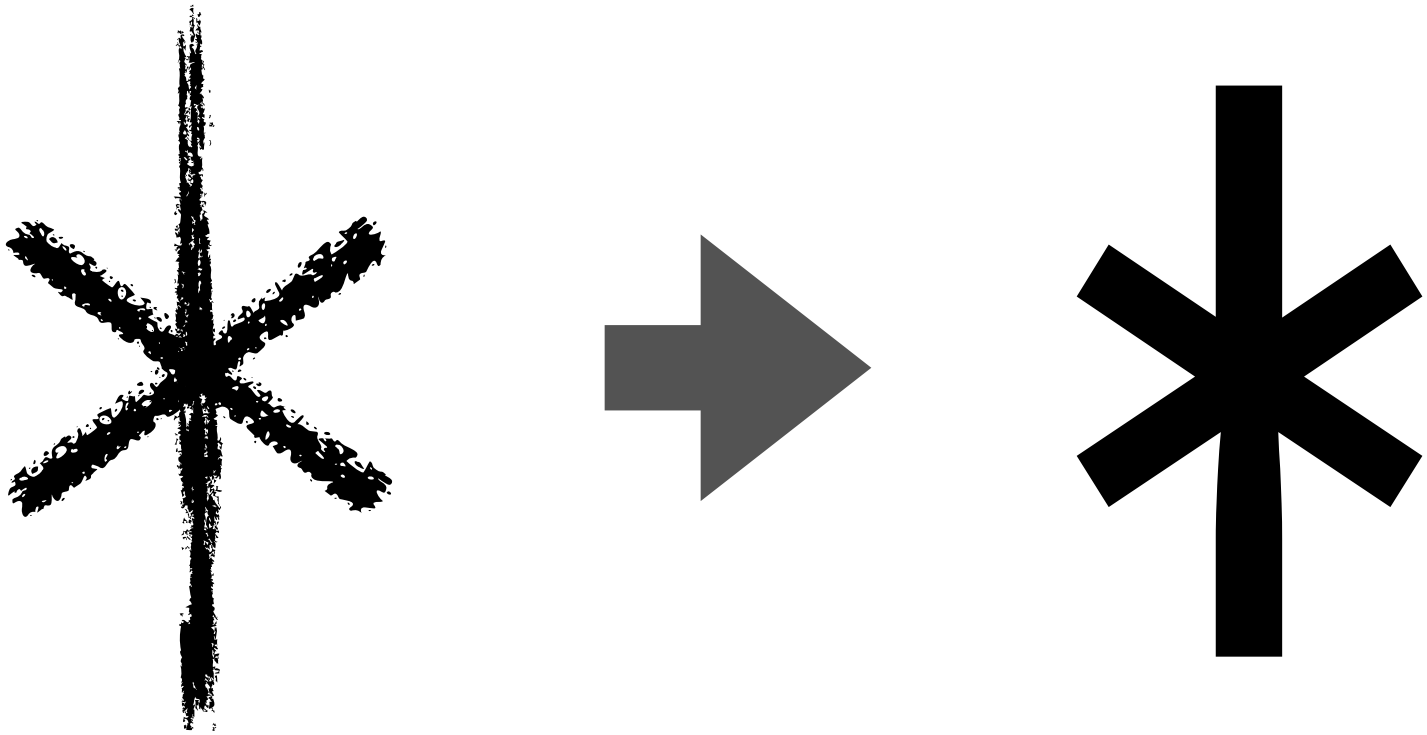


Latin

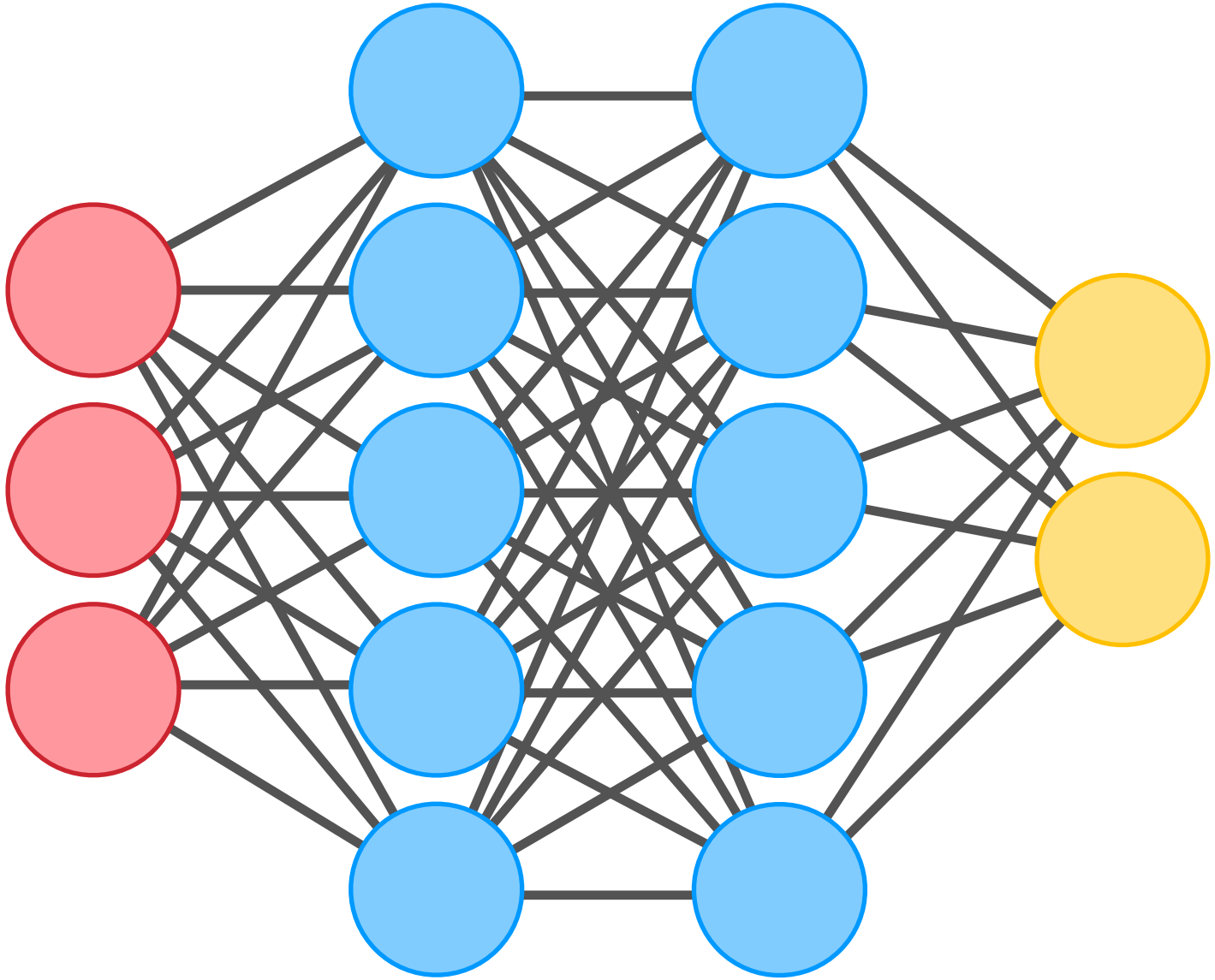


Modern Latin

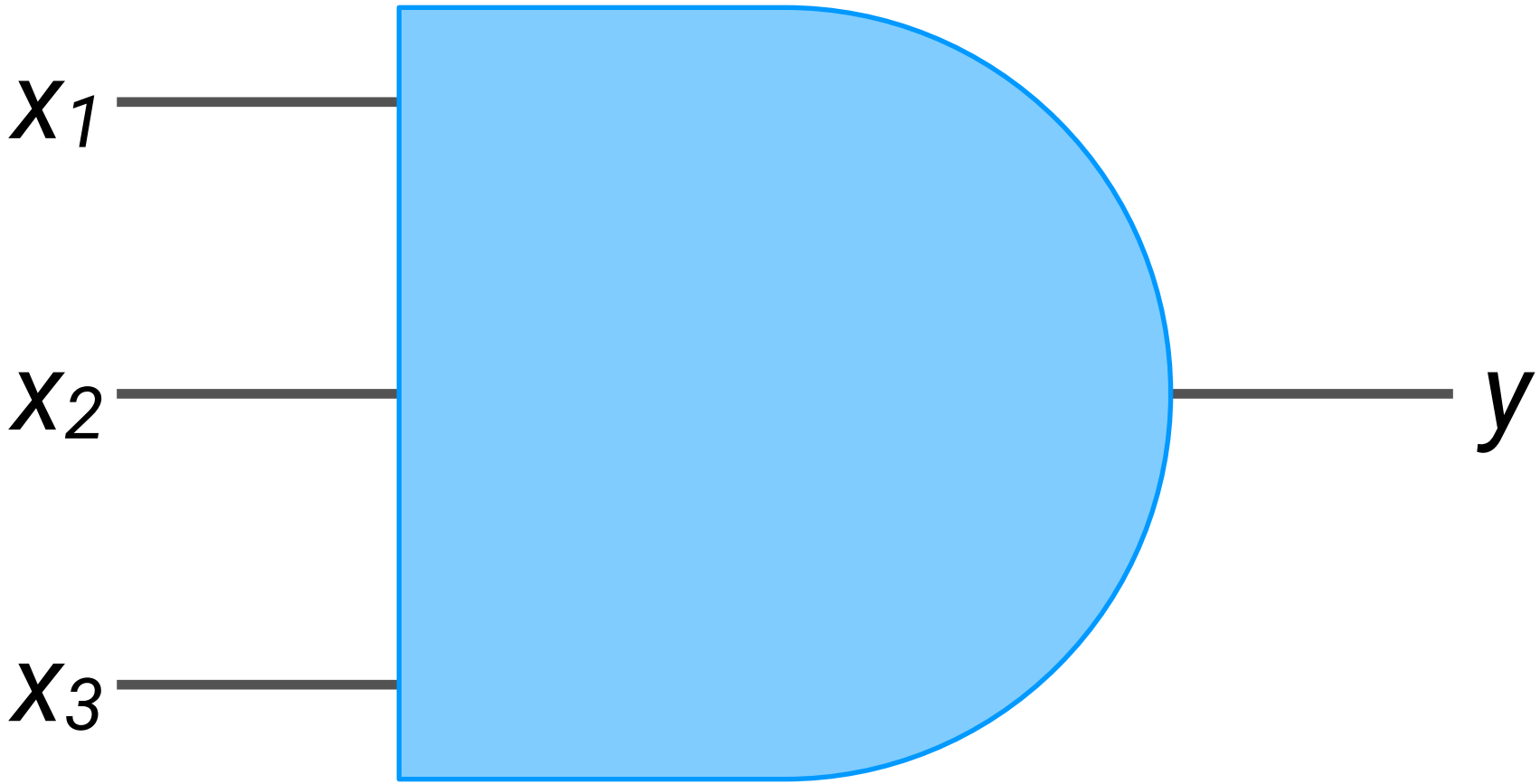
Recognizing Runes



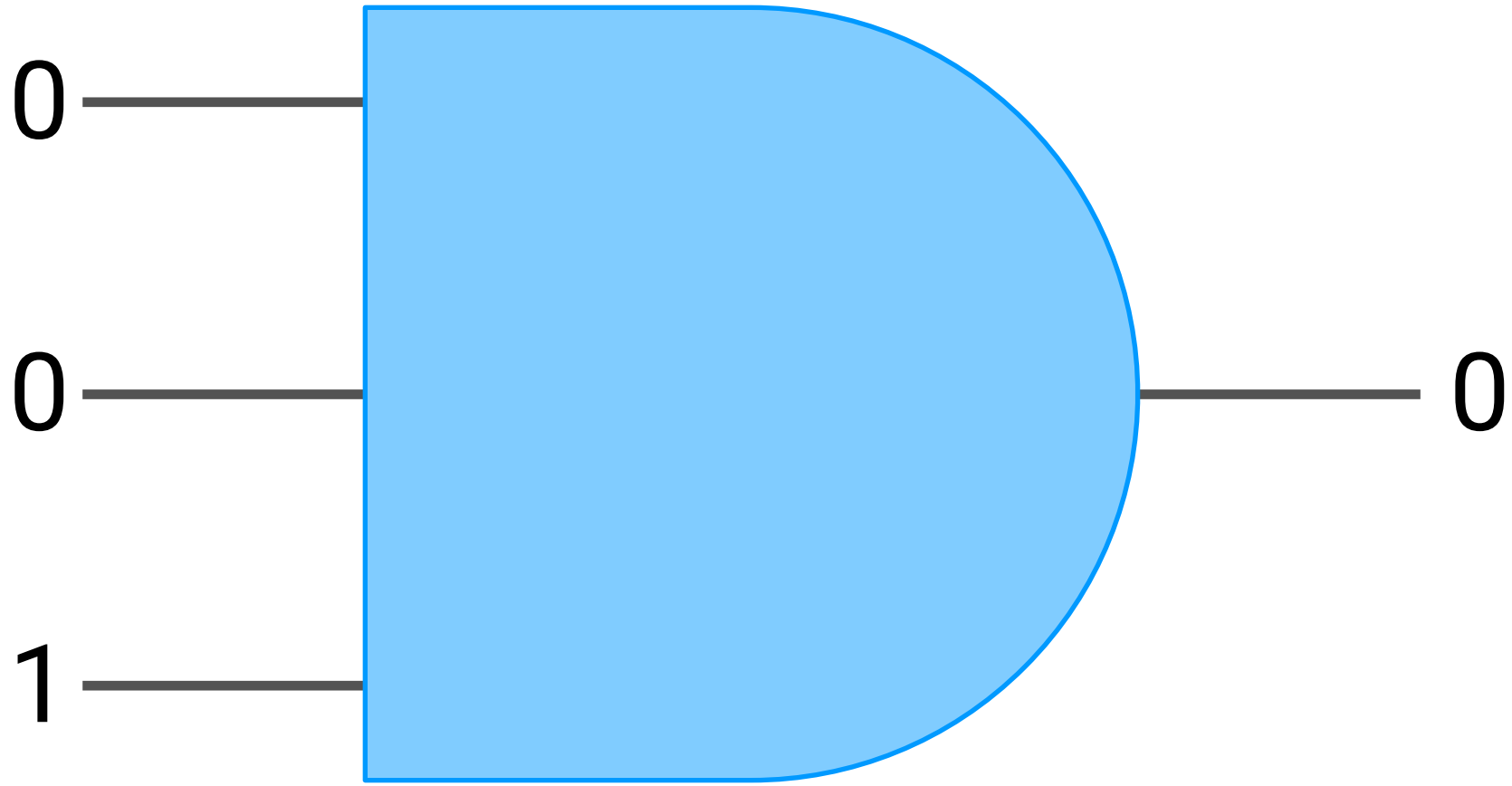
Neural Networks



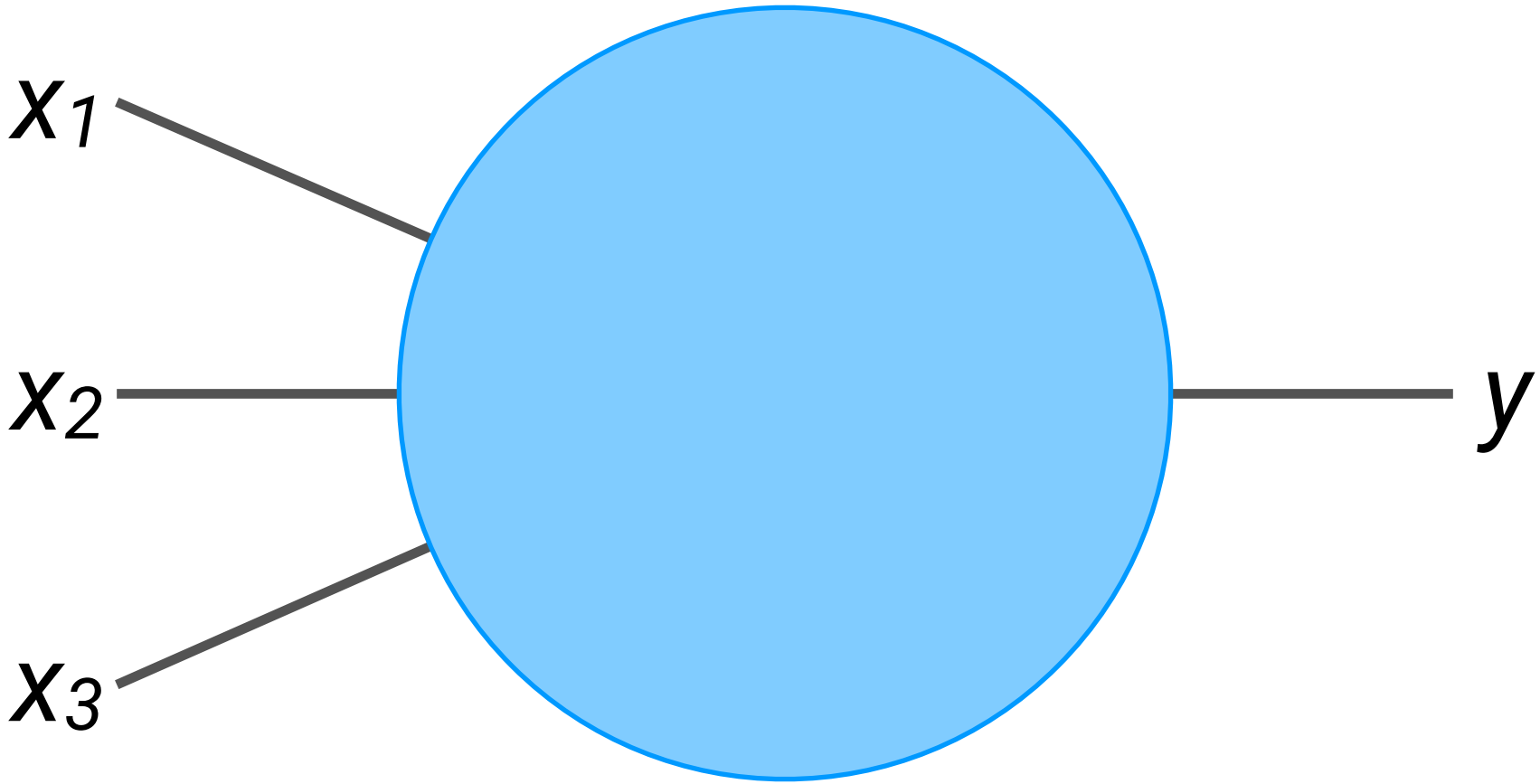
Logic Gates



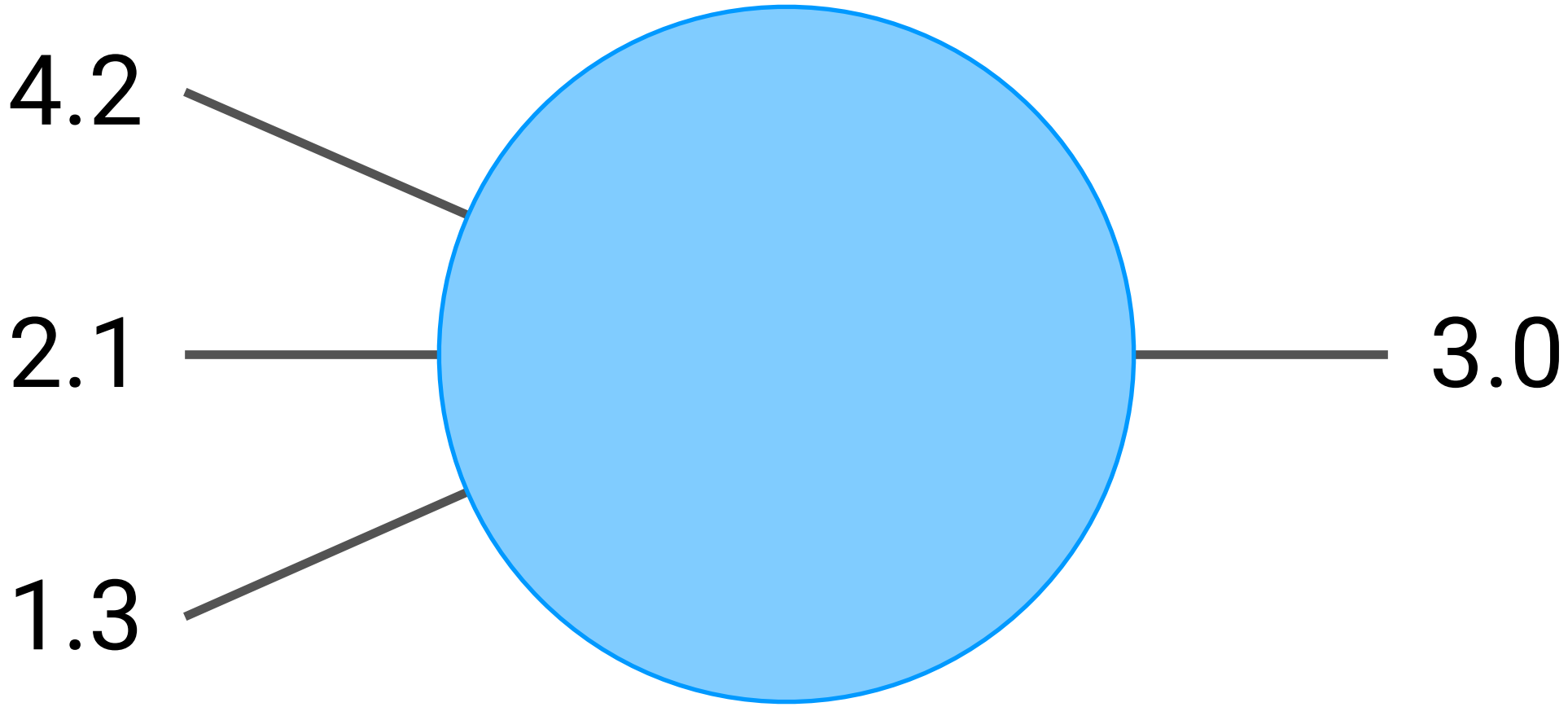
Logic Gates with Actual Numbers



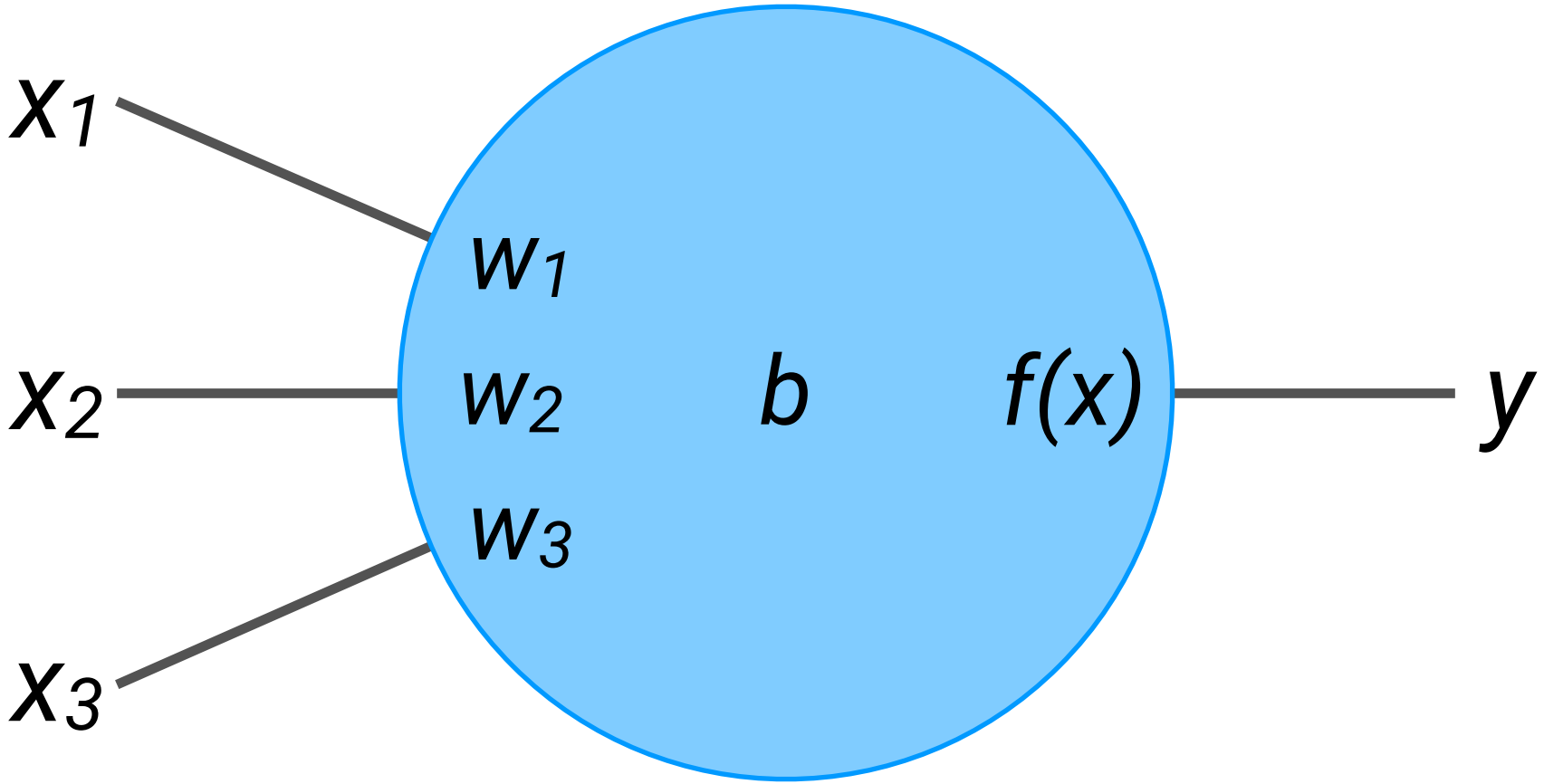
Neurons



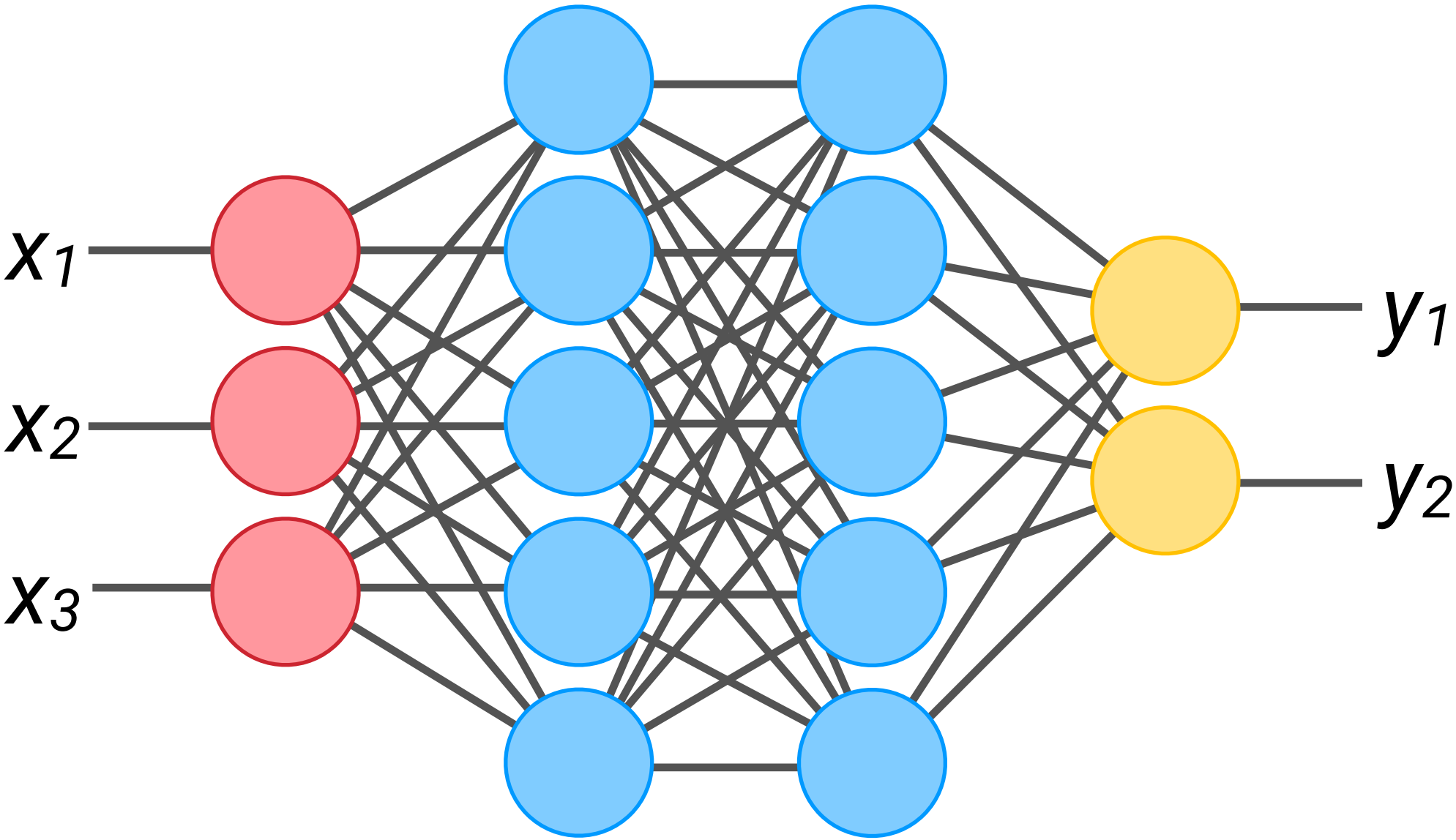
Neurons with Actual Numbers



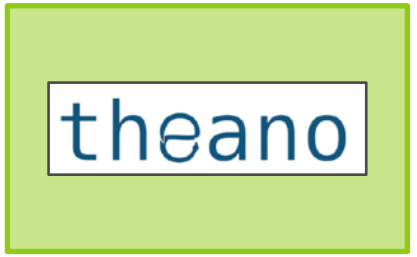
Inside a Neuron



Neural Networks



Keras



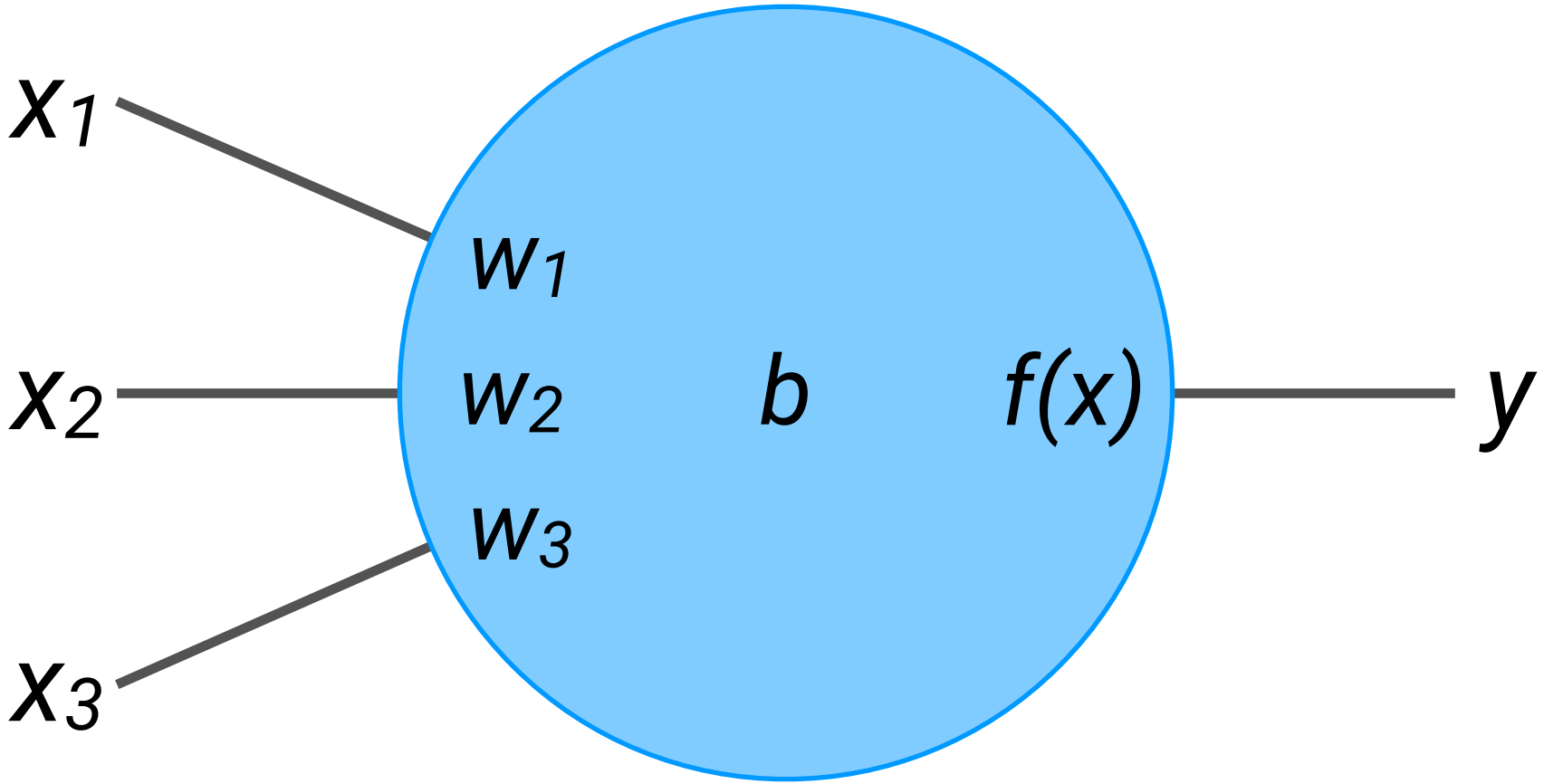
```
# configure the neural network
model = Sequential()
model.add(Dense(24, input_shape=(2, ), activation='relu'))
model.add(Dense(2, activation='softmax'))

# compile the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# train it
model.fit(X_train, Y_train, batch_size=32, epochs=5, verbose=1)
```

Exercise 1

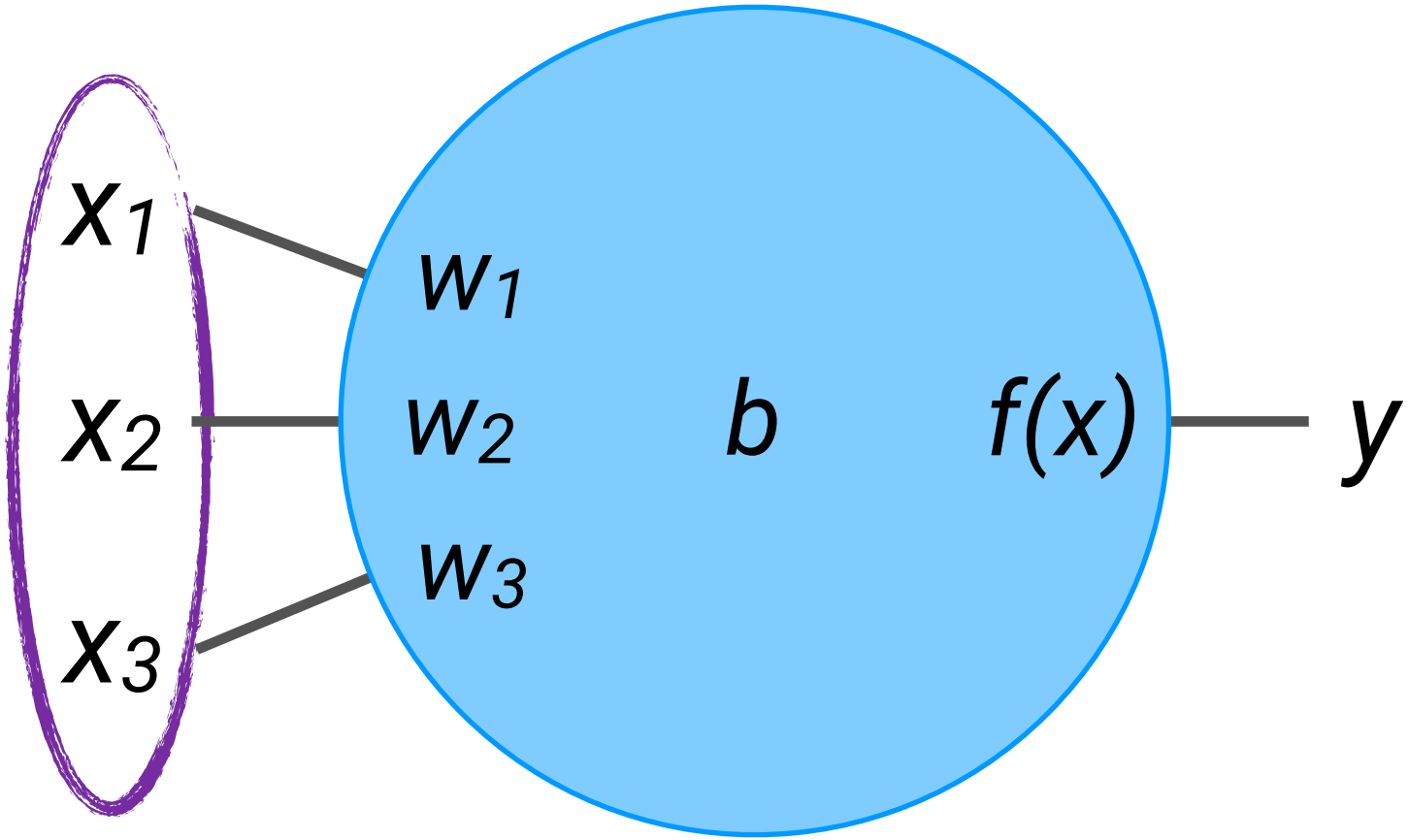
Logic Gates the Hard Way

https://github.com/guyroyse/deep-learning-like-a-viking

Inside a Neuron

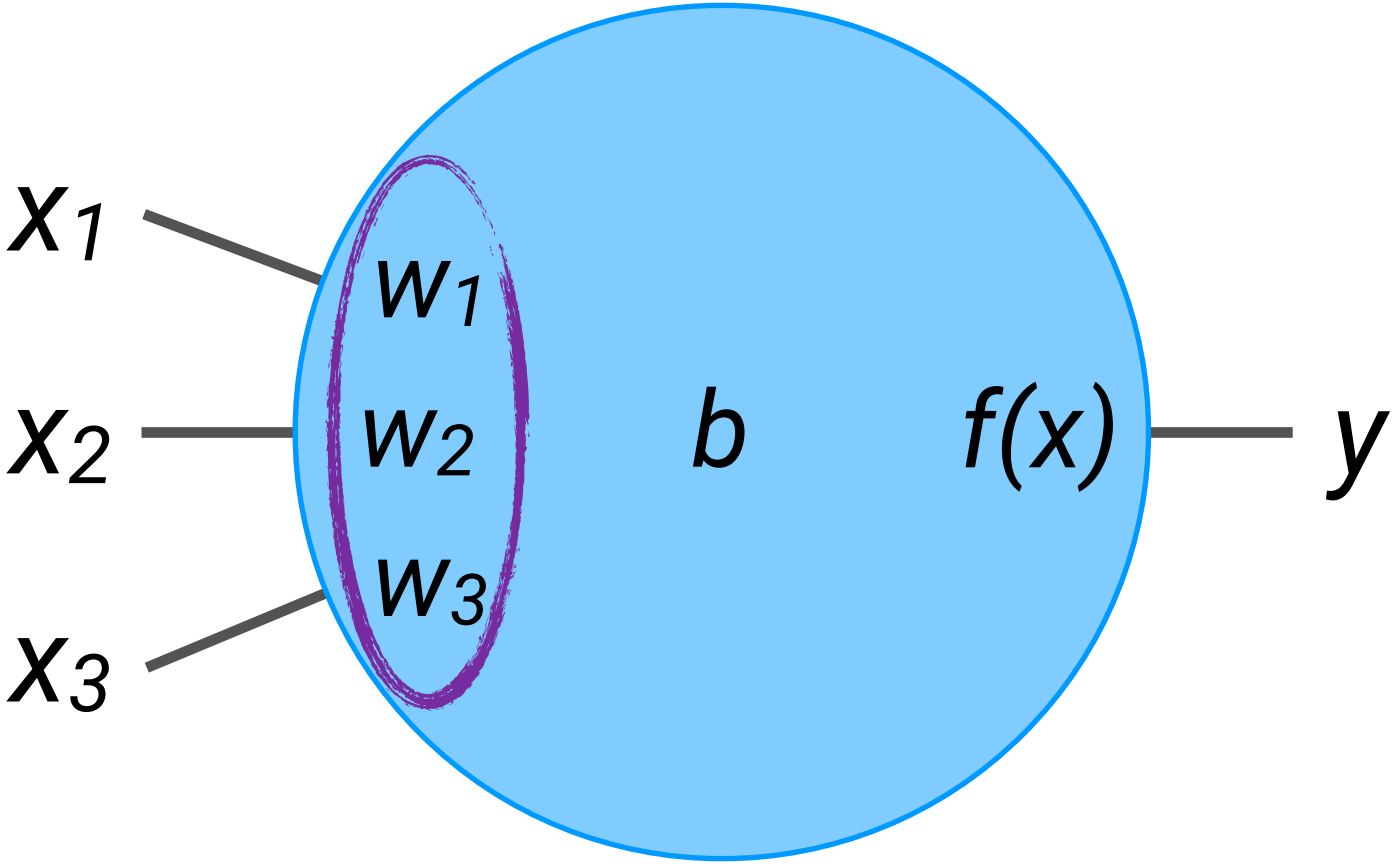


Inputs



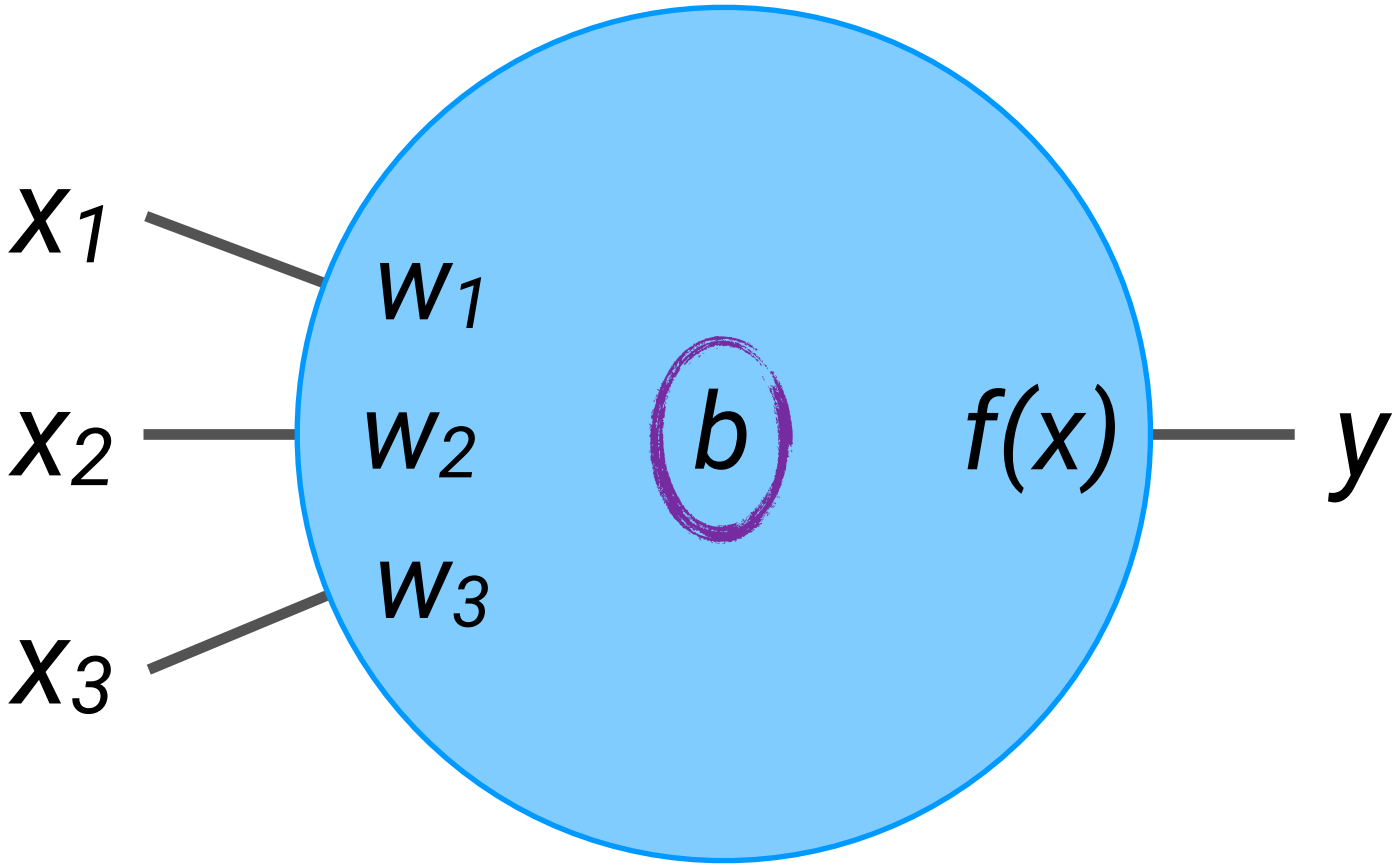
The values coming into the neuron.

Weights



Makes an input more or less important.

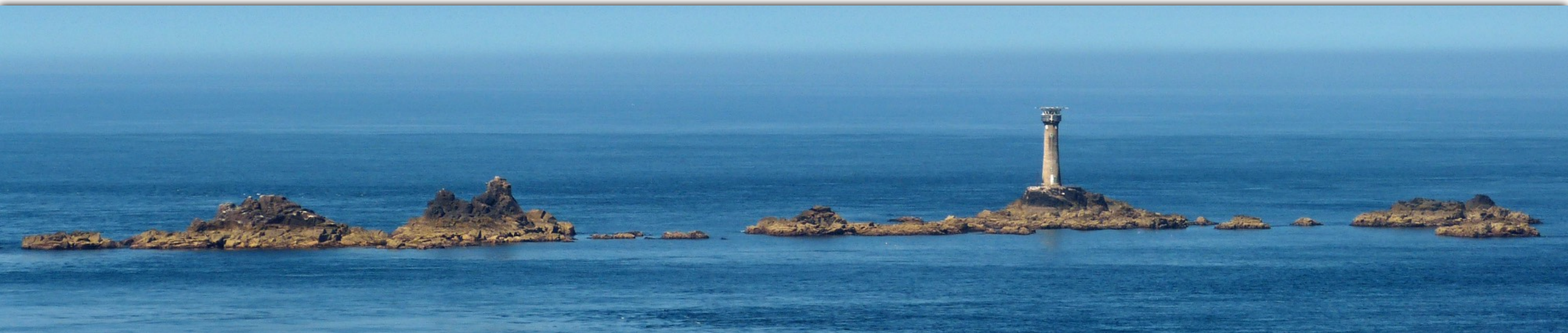
Bias



Makes the sum of the inputs more or less important.

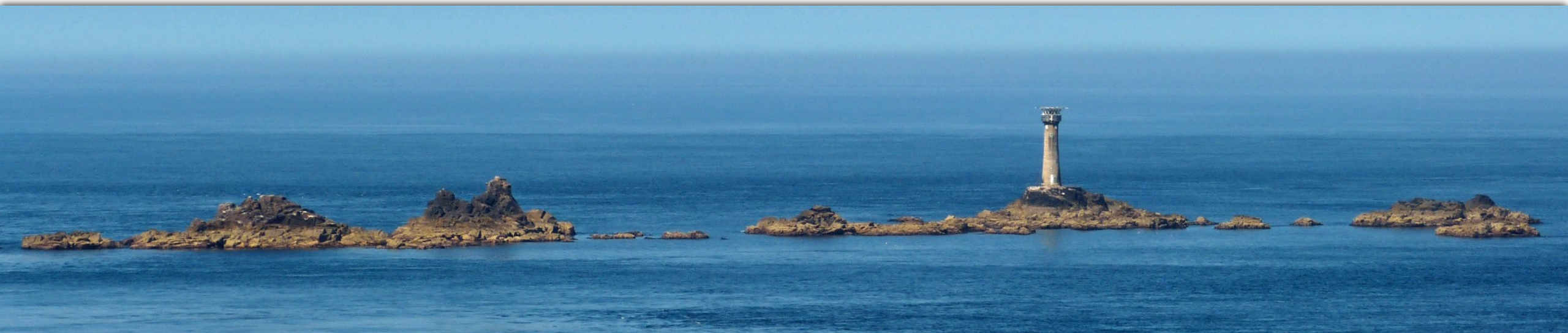
The Math So Far

$$x_1 w_1 + x_2 w_2 + x_3 w_3 + b$$

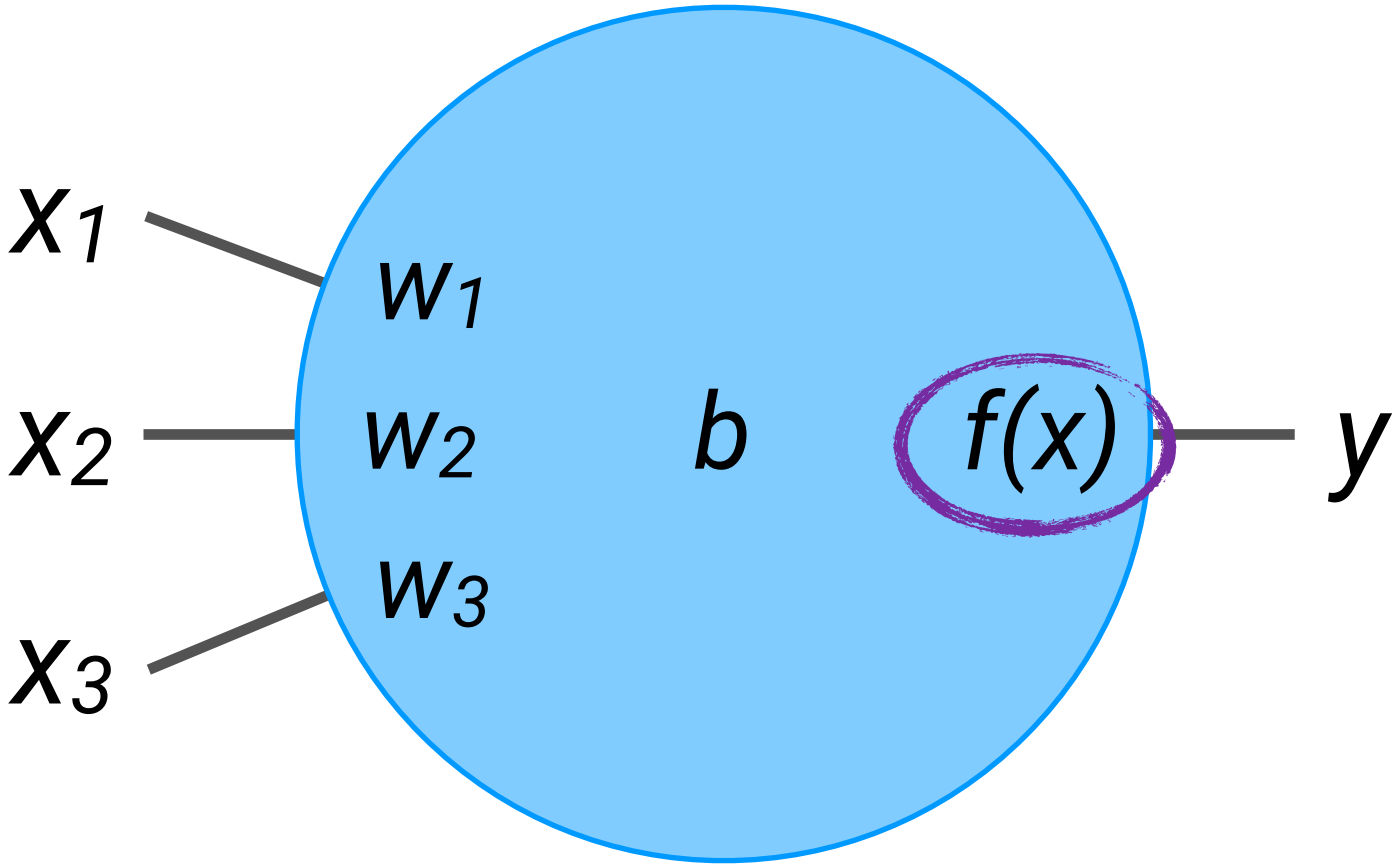


Even Mathier

$$\Sigma_i x_i w_i + b$$

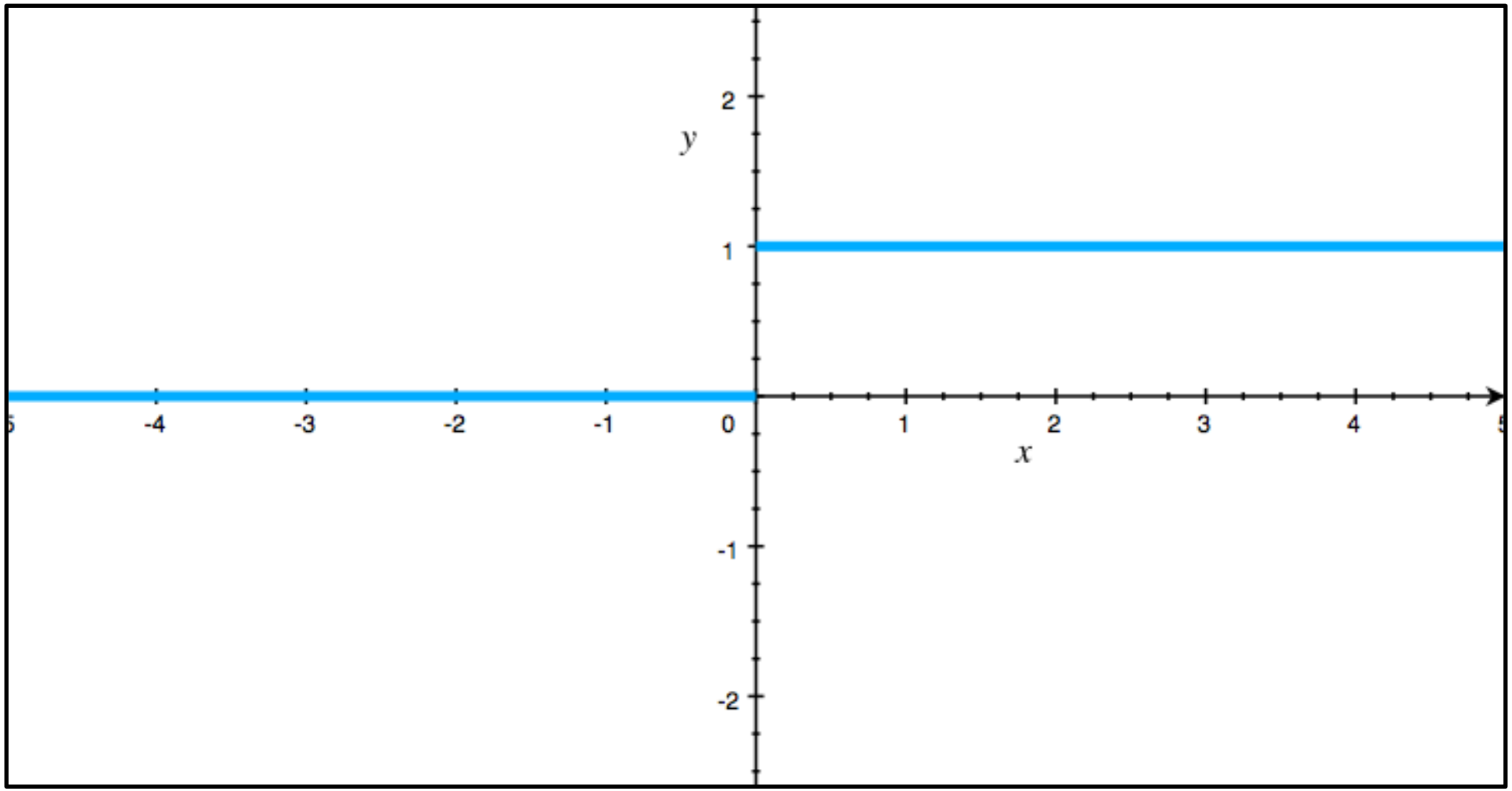


Activation Function

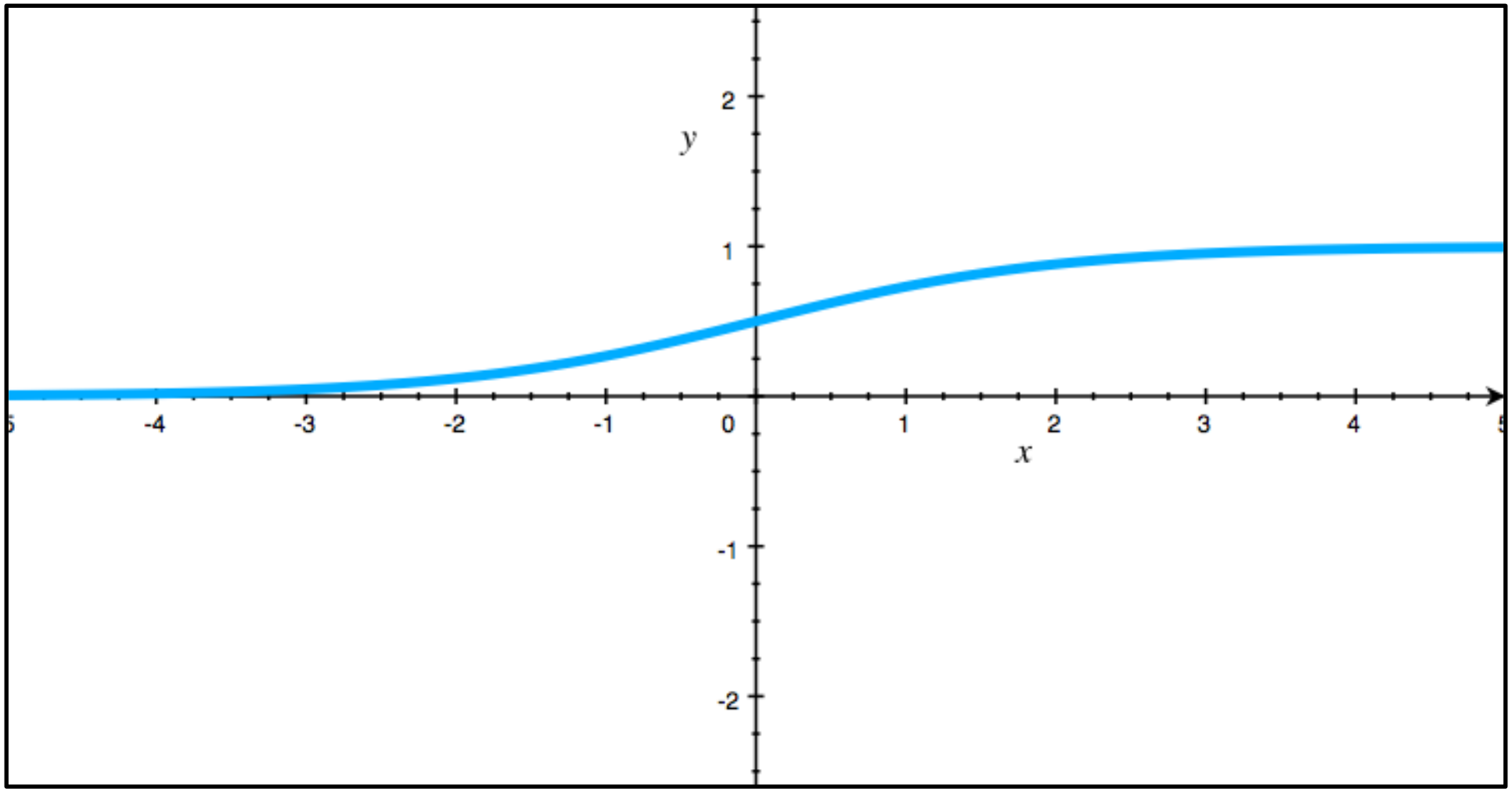


Adjusts the output of the neuron.

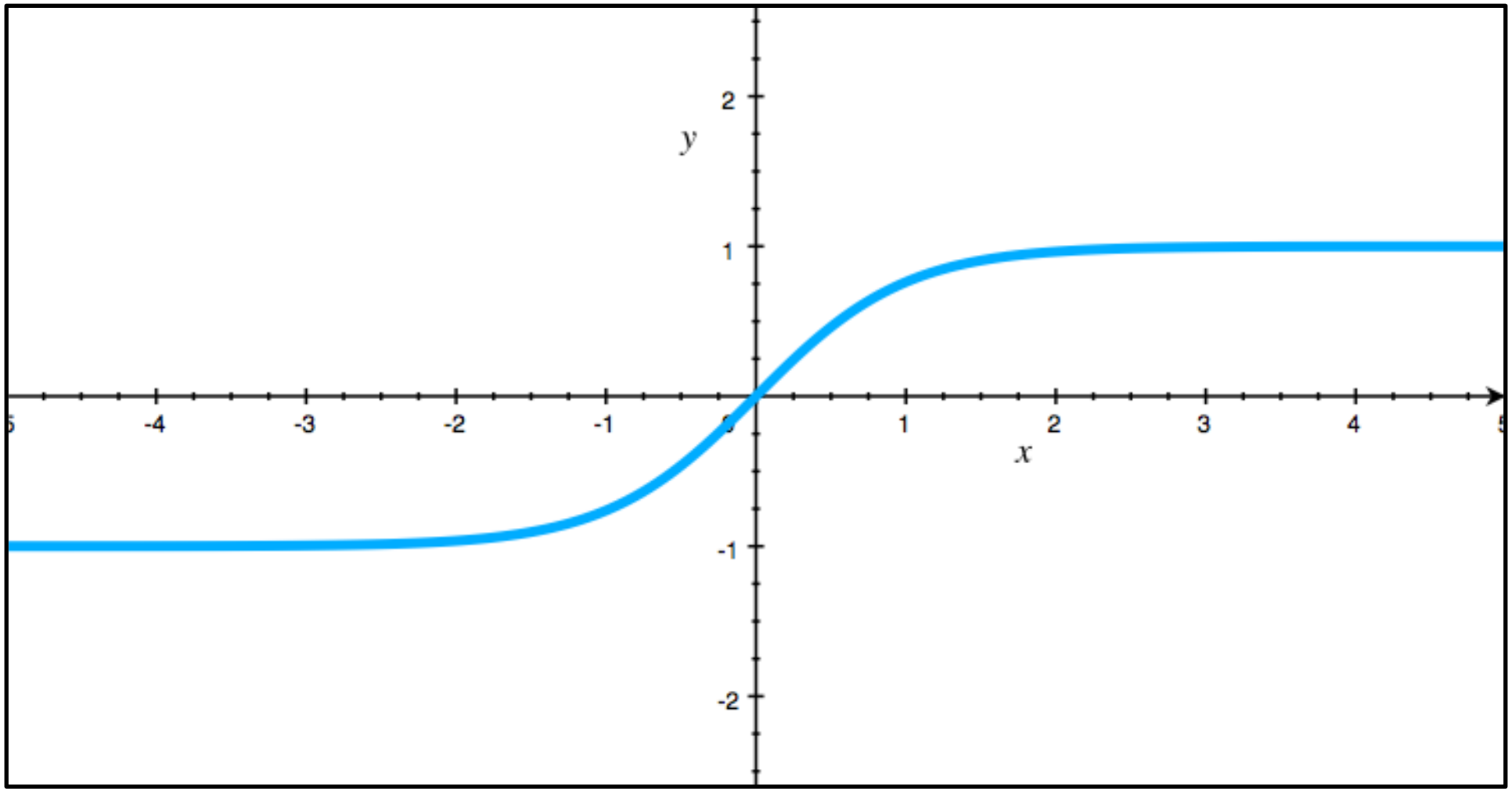
Step Function



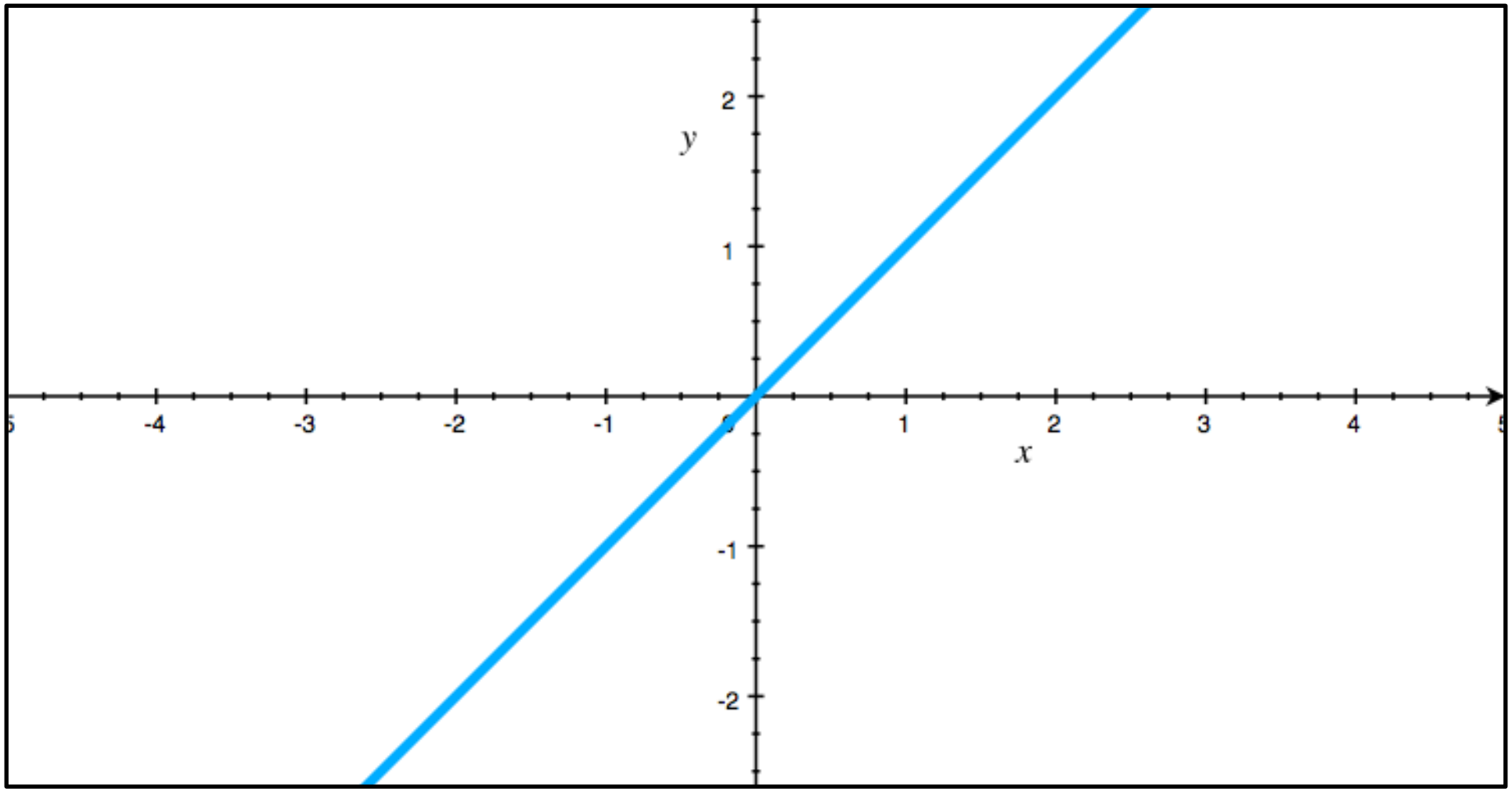
Sigmoid Function



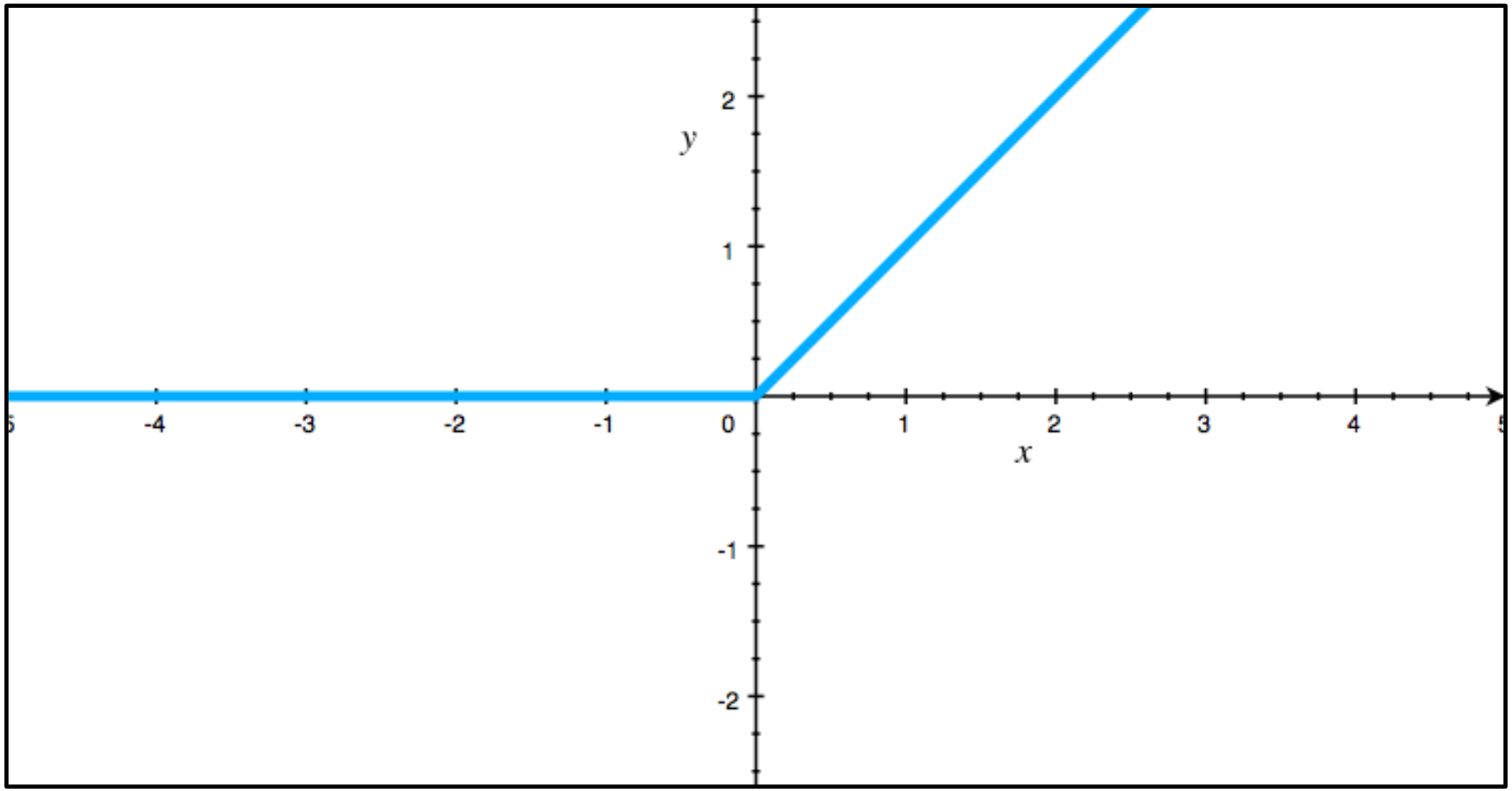
tanh Function



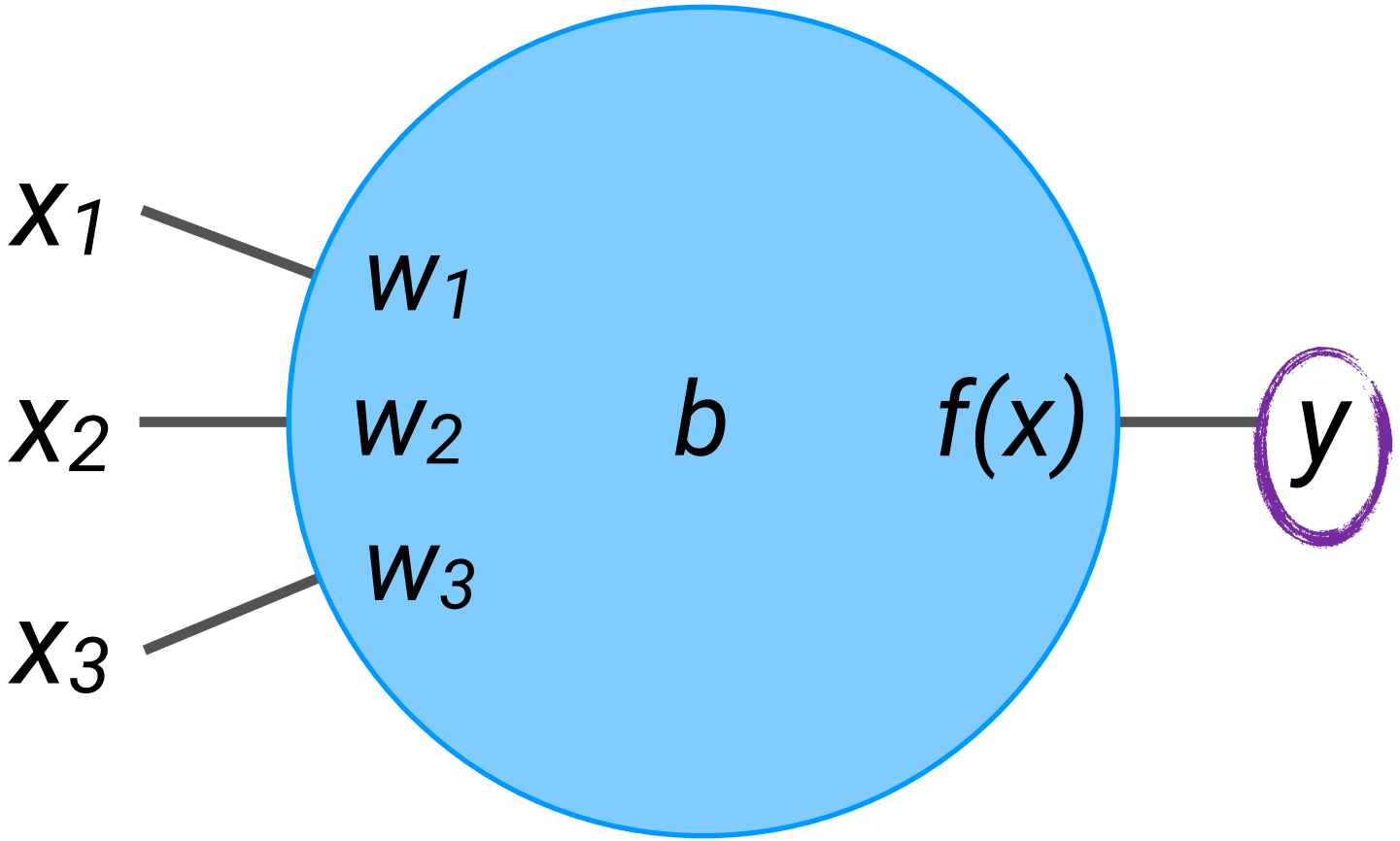
Linear Function



Rectified Linear Units (ReLU)

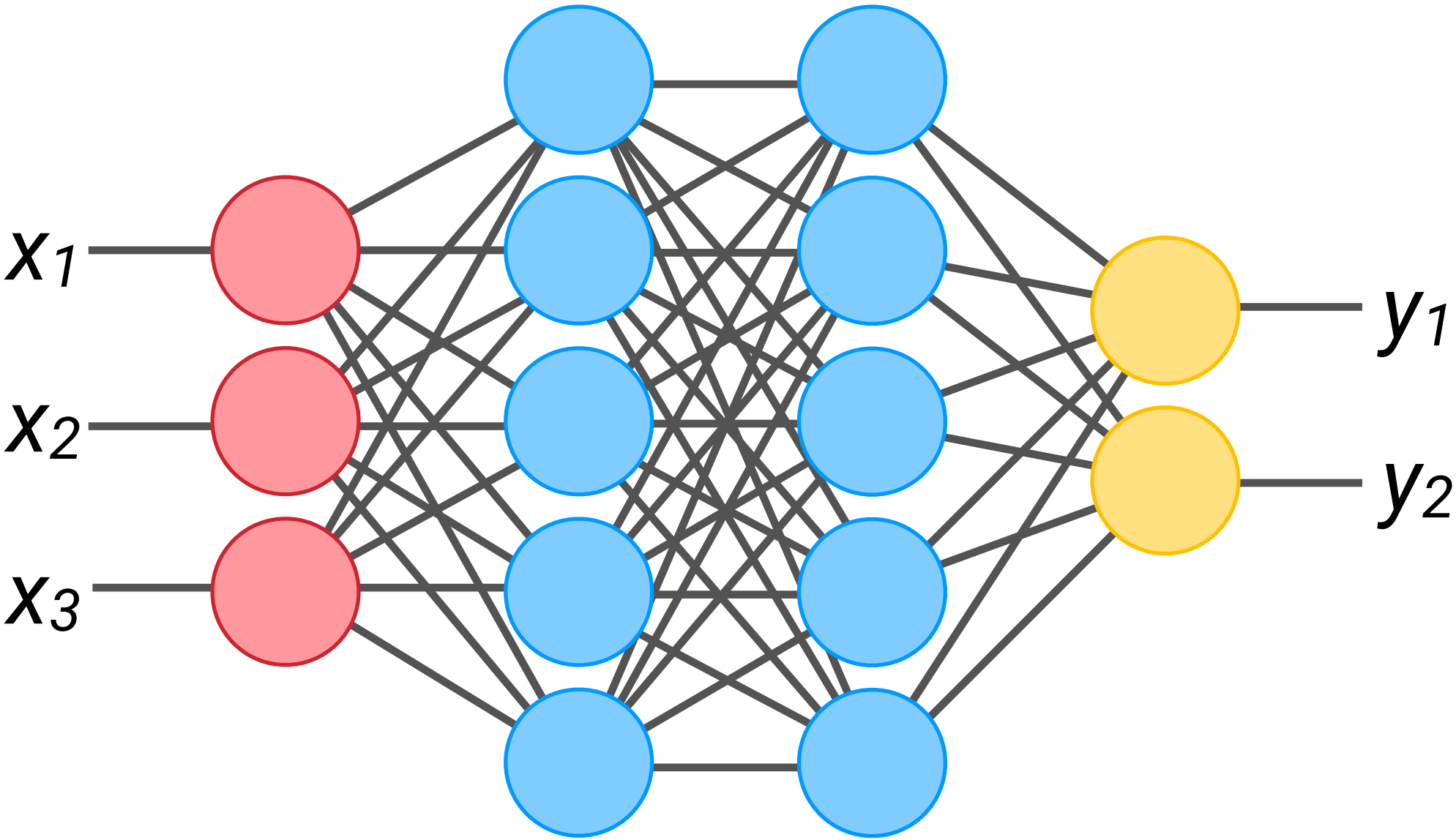


Output



The value leaving the neuron.

Neural Networks

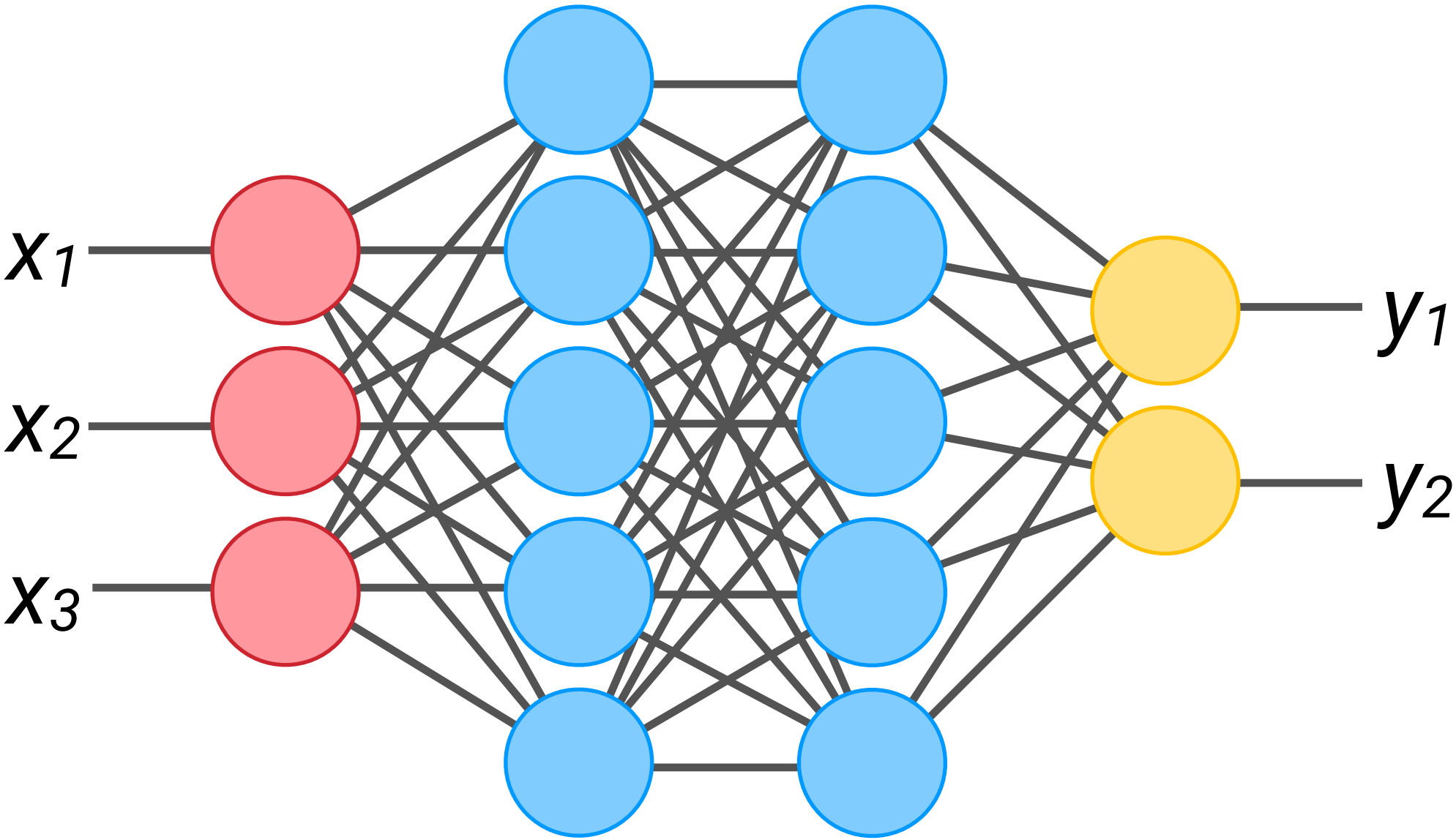


```
# configure the neural network
model = Sequential()
model.add(Dense(128, input_shape=(1, 24, 24), activation='relu'))
model.add(Flatten())
model.add(Dense(16, activation='softmax'))

# compile the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# train it
model.fit(X_train, Y_train, batch_size=32, epochs=20, verbose=1)
```

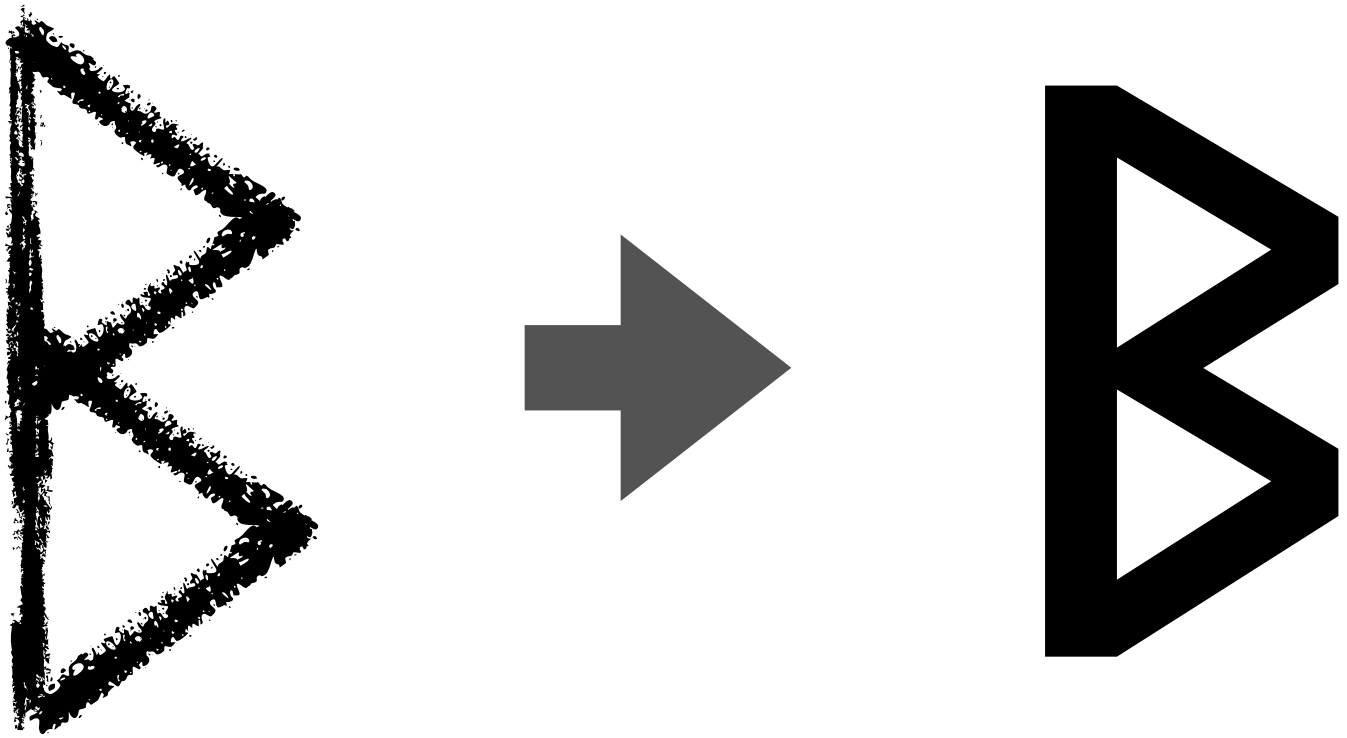
Exercise 2

Recognizing Runes

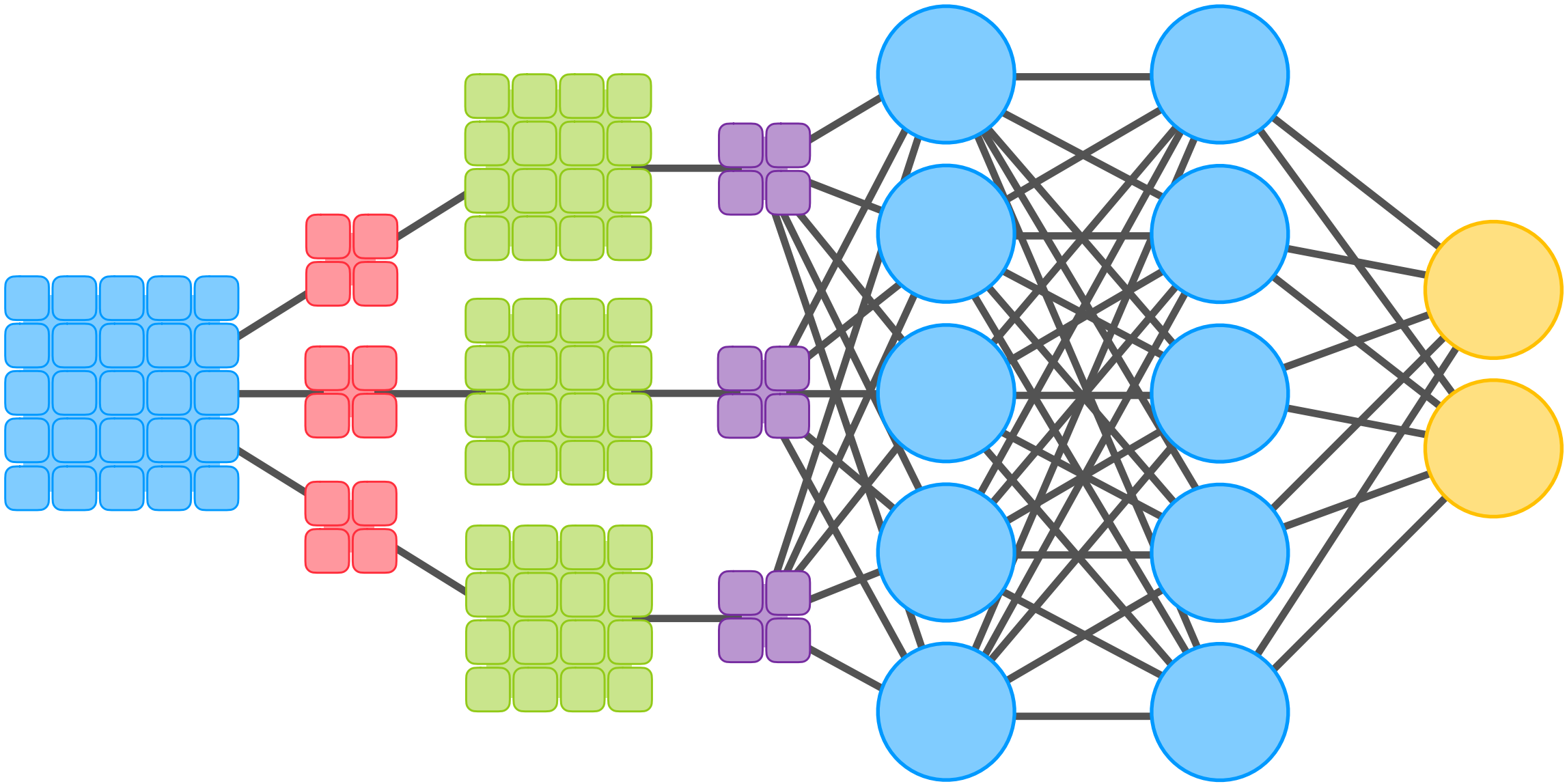
Neural Networks



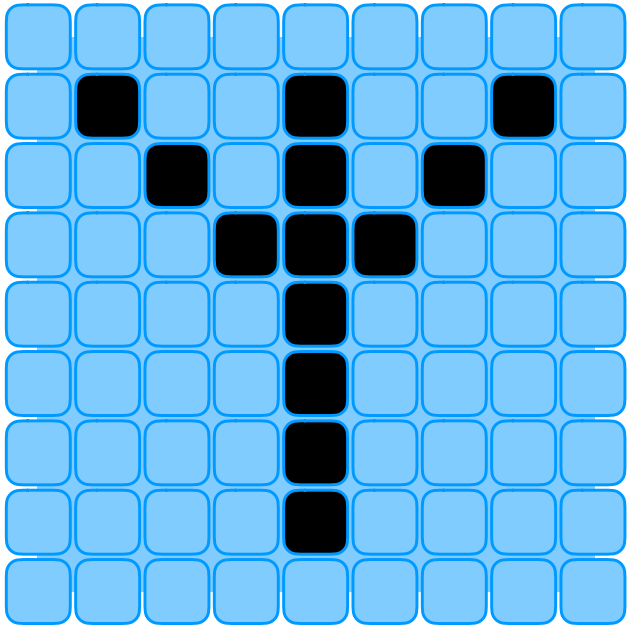
Bad at Recognizing Runes



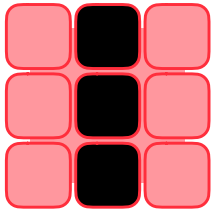
Convolutional Neural Networks



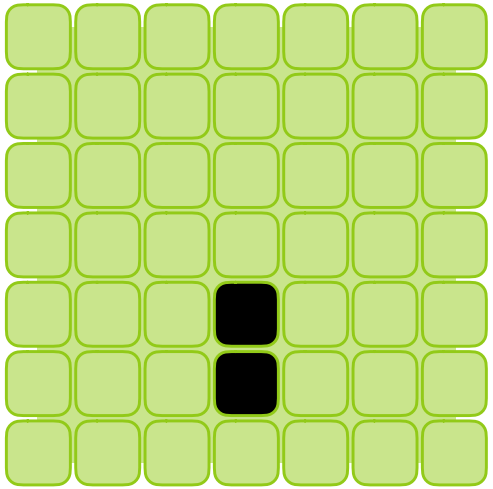
Convolutions

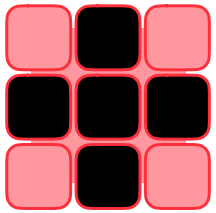


Input

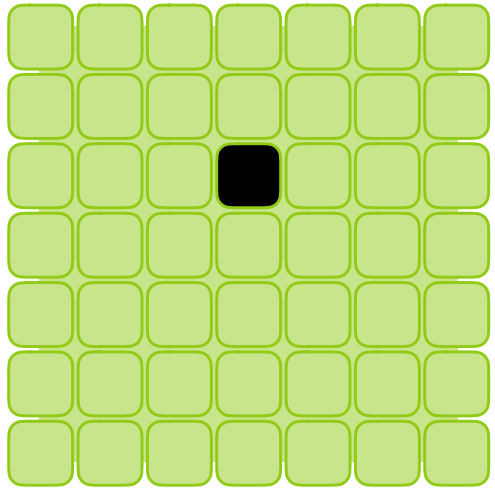


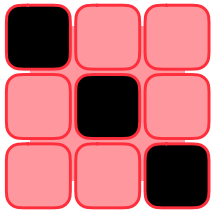
Filter



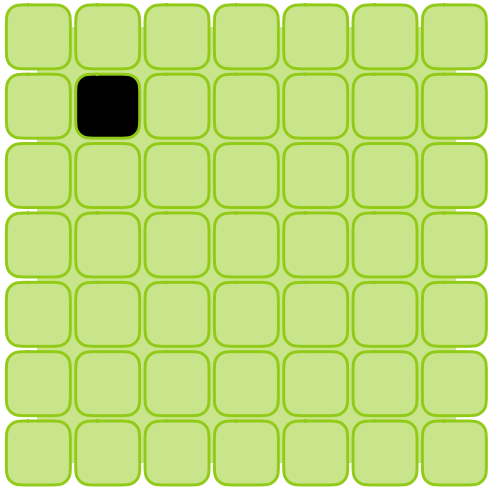


Filter

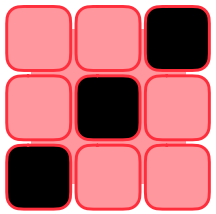




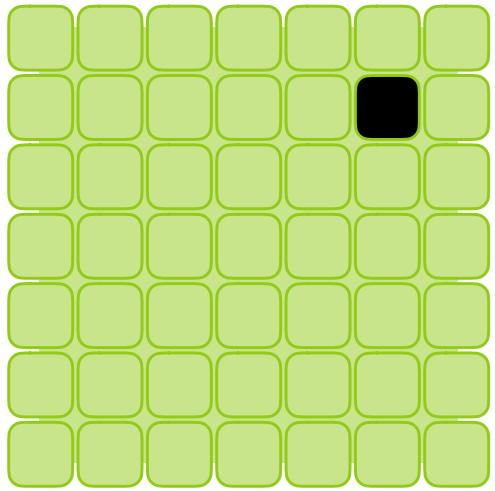
Filter



Output

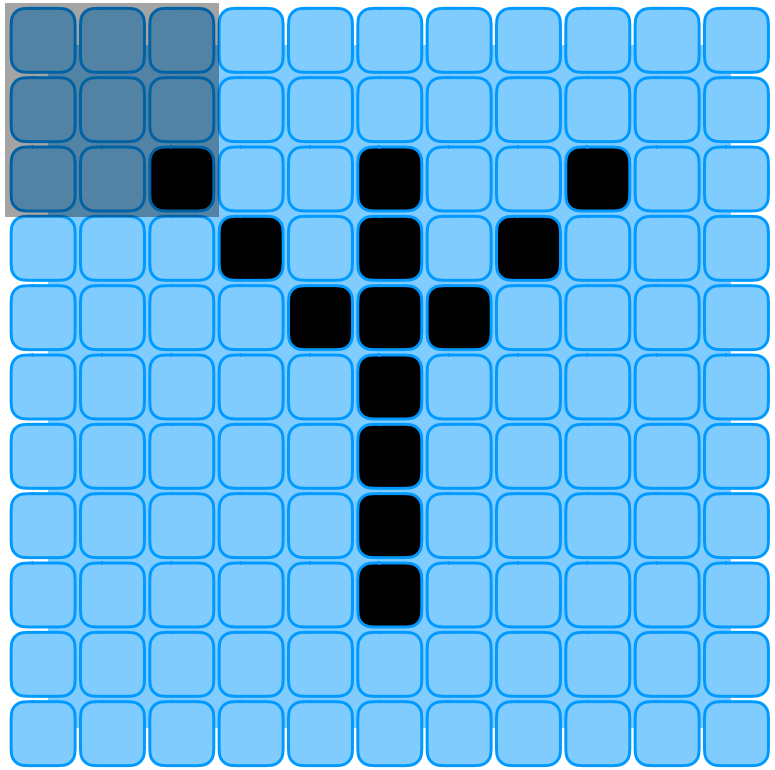


Filter

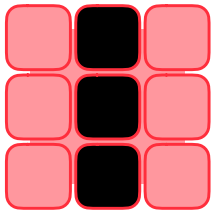


Output

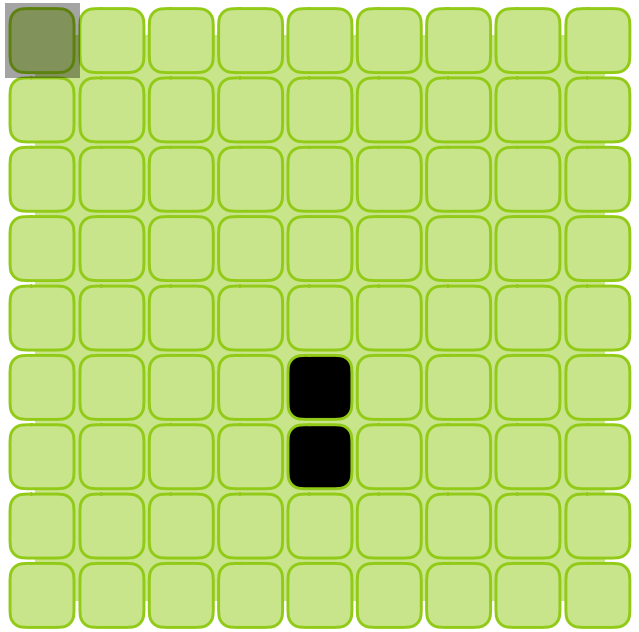
Convolutions with Fancy Animation



Input

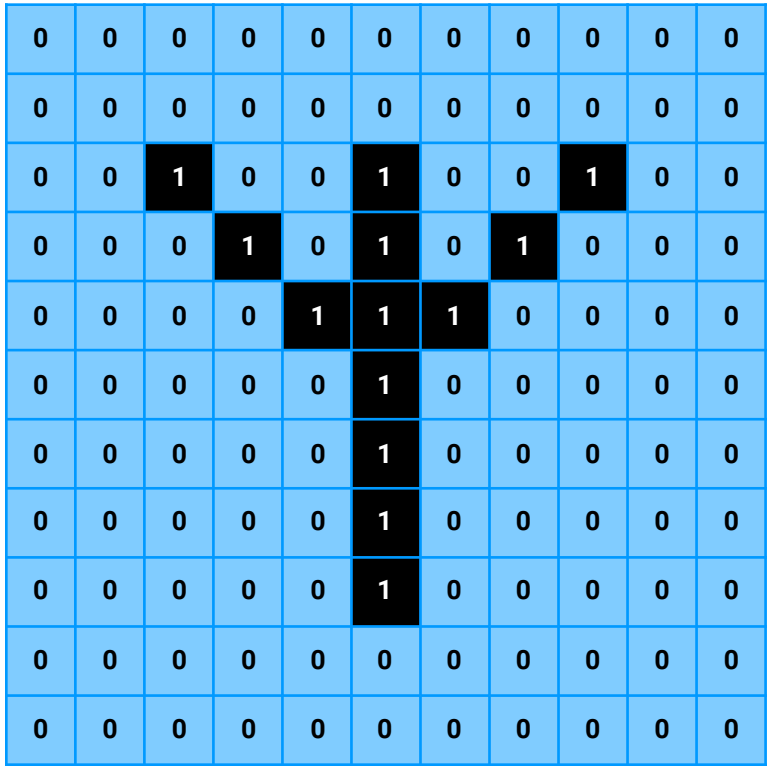


Filter

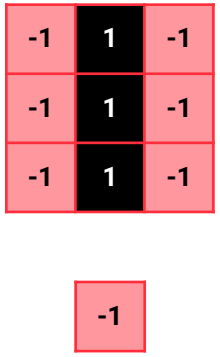


Output

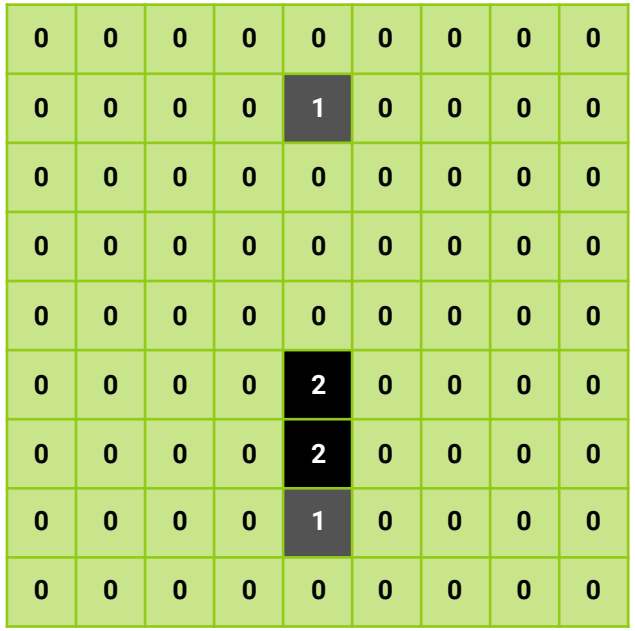
Convolutions with Numbers



Input



Filter



Output

Convolutions with Familiar Looking Numbers

x_i

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	1	0	0
0	0	0	1	0	1	0	1	0	0	0
0	0	0	0	1	1	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

w_i

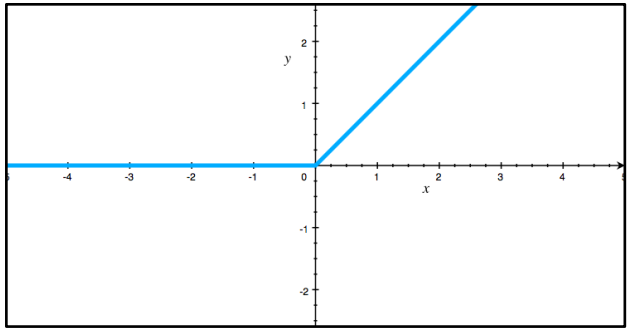
-1	1	-1
-1	1	-1
-1	1	-1

-1

b

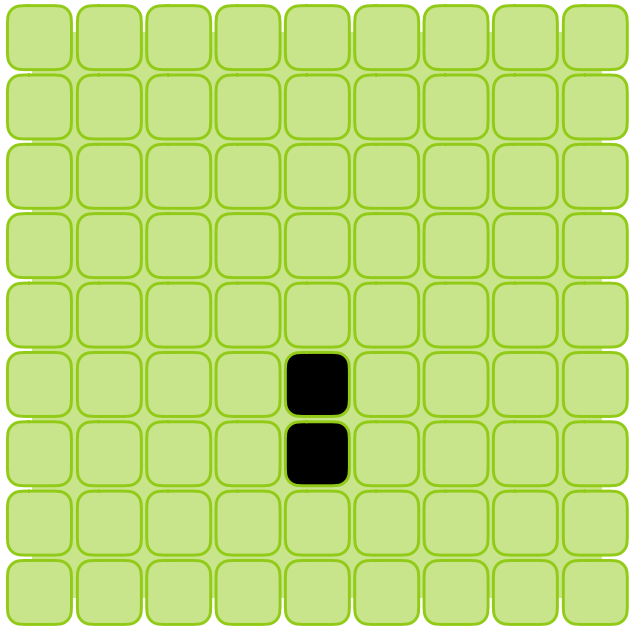
y_i

0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0	0
0	0	0	0	2	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0

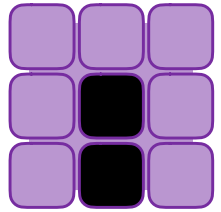


$$\Sigma_i \, x_i w_i + b$$

Pooling



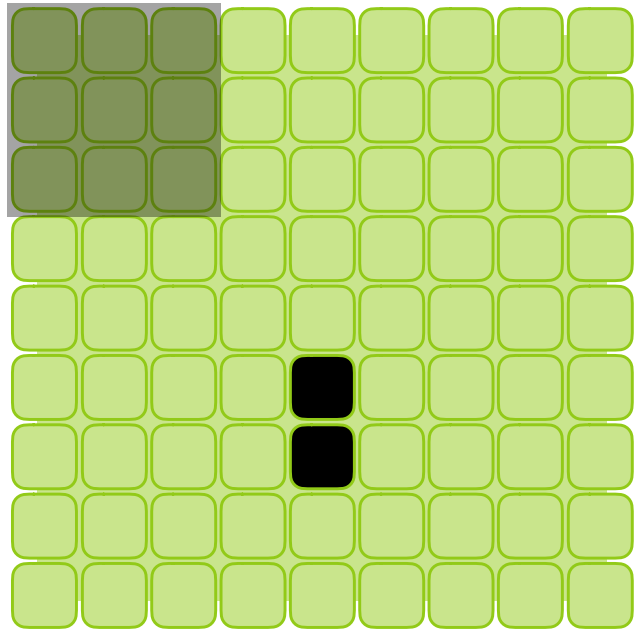
3 x 3



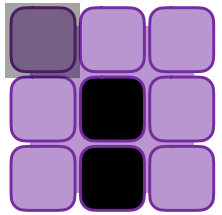
Output

Pooled

Pooling with Fancy Animation



3 x 3



Output

Pooled

Pooling with Numbers

0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0	0
0	0	0	0	2	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0

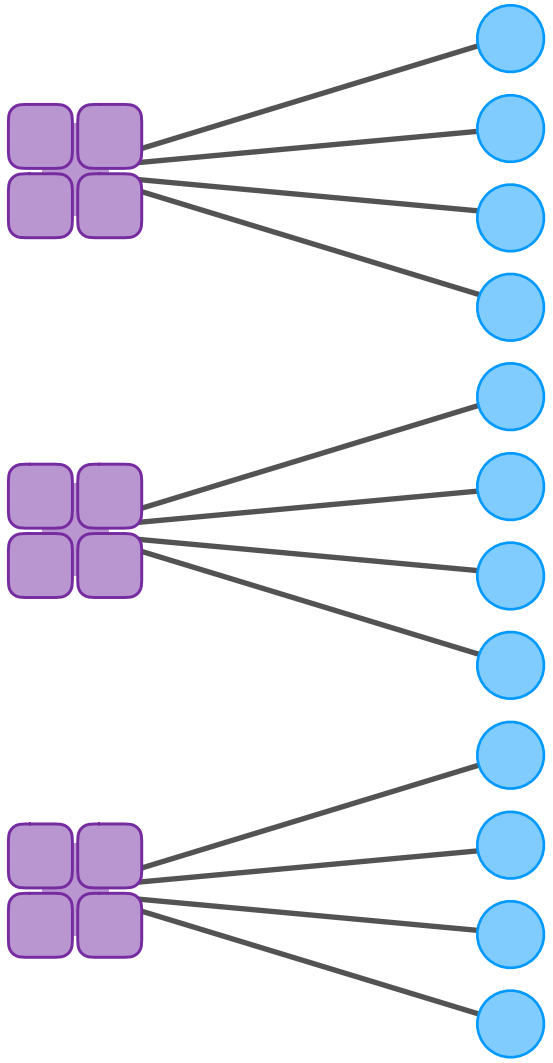
Output

3 x 3

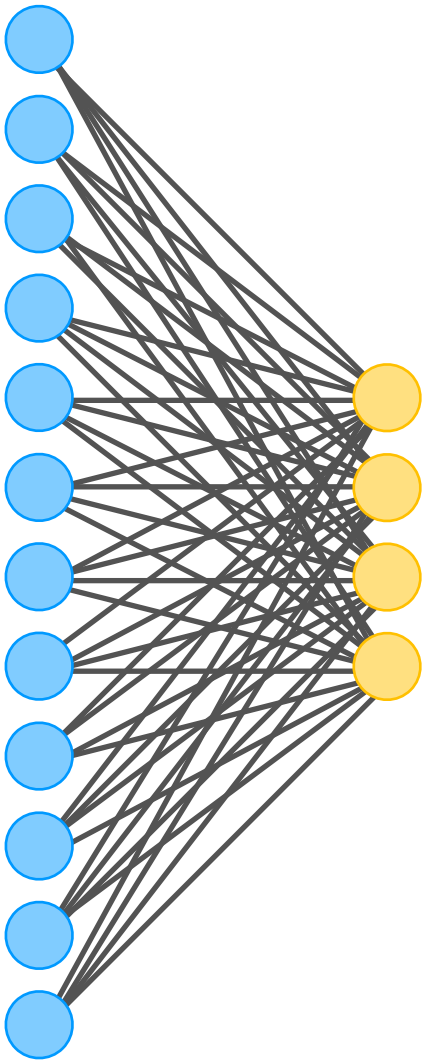
0	1	0
0	2	0
0	2	0

Pooled

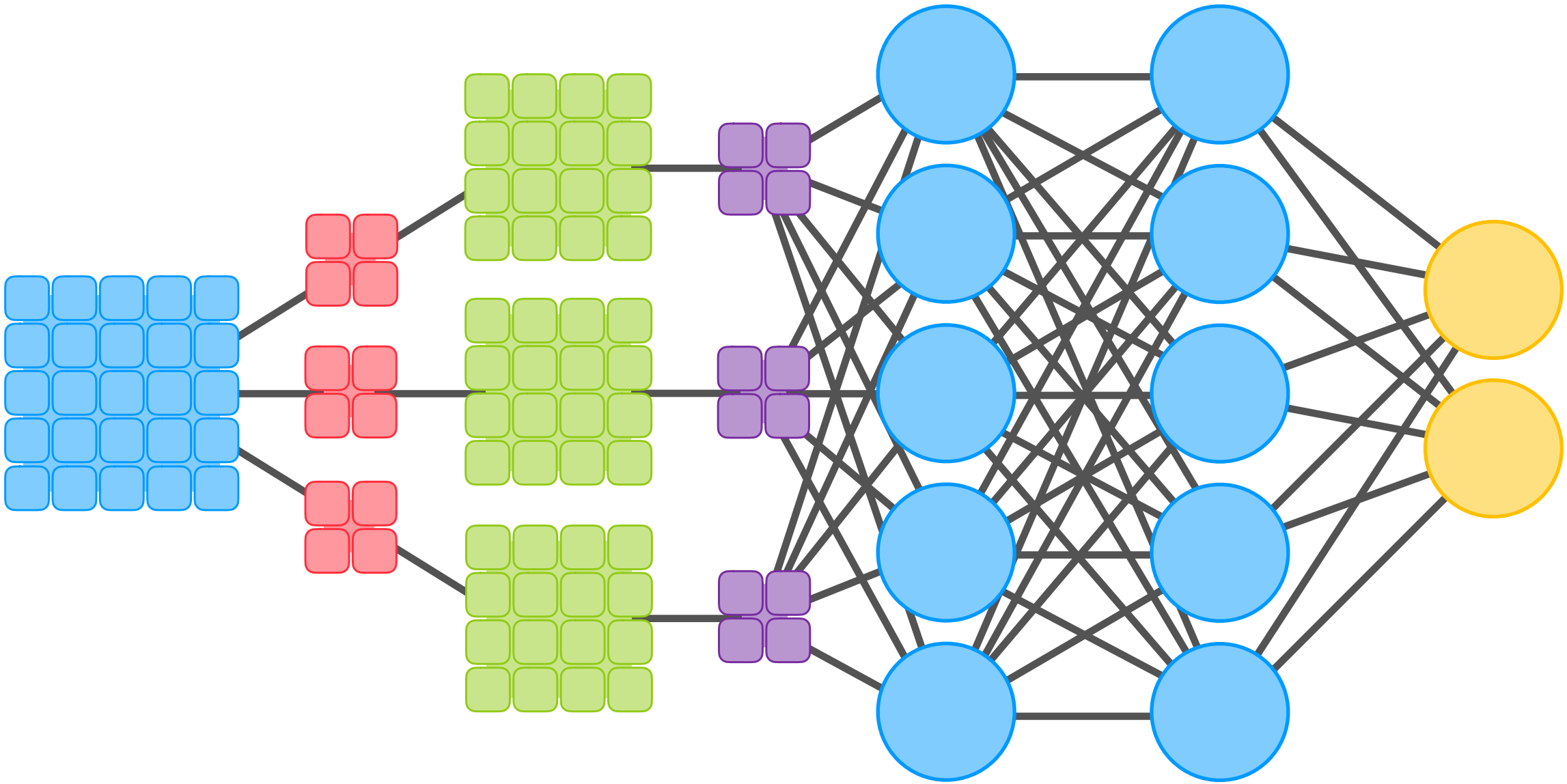
Flattening



Fully Connected



Convolutional Neural Networks



```
# configure the neural network
model = Sequential()
model.add(Conv2D(48, (3, 3), activation='relu', data_format='channels_first', input_shape=(1, 24, 24)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(24, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(16, activation='softmax'))

# compile the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# train it
model.fit(X_train, Y_train, batch_size=32, epochs=20, verbose=1)
```

Exercise 3

Recognizing Runes the Cool Way

Exercise 4

Put It In An App

Resources

Keras
https://keras.io/

Microsoft CNTK
https://www.microsoft.com/en-us/cognitive-toolkit/

TensorFlow
https://www.tensorflow.org/

Theano
http://www.deeplearning.net/software/theano/

Flask
http://flask.pocoo.org/

VanillaJS
http://vanilla-js.com/

Younger Futhark
https://en.wikipedia.org/wiki/Younger_Futhark

Neural Network Zoo
http://www.asimovinstitute.org/neural-network-zoo/

A Beginner's Guide to to Neural Networks
https://towardsdatascience.com/a-beginners-guide-to-neural-networks-b6be0d442fa4

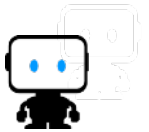
Hacker's Guide to Neural Networks
http://karpathy.github.io/neuralnets/

CS231n Convolutional Neural Networks for Visual Recognition
http://cs231n.github.io/convolutional-networks/

An Intuitive Guide to Convolutional Neural Networks
https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050

https://github.com/guyroyse/deep-learning-like-a-viking

Image Credits



- https://www.flickr.com/photos/frankdouwes/3985117642/
- https://www.flickr.com/photos/ecastro/4415693080/
- https://www.flickr.com/photos/torsven/2869528719/
- https://www.flickr.com/photos/arg_flickr/14732931742/
- https://www.flickr.com/photos/wwarby/23841229208/
- https://www.flickr.com/photos/hesim/6553273471/

Guy Royse

Developer Evangelist

DataRobot

guyroyse

code.guy.dev

guy.dev


DataRobot