Machine Learning in Games

Adam Sattaur, Sam Devlin, Hendrik Baier

Machine learning in games

- Move prediction
- Win prediction

- Churn prediction
- First purchase prediction

Example application: Spades

- Move prediction
- Human gameplay

Supervised learning

- Learning from examples

Supervised learning

- Learning from examples
- e.g. “Cat or Carlos?”

“Cat”

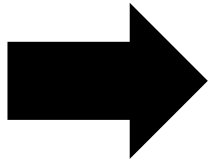
“Cat”

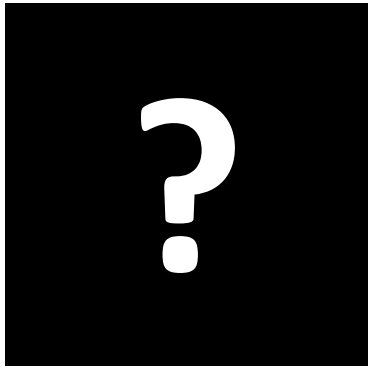
“Carlos”

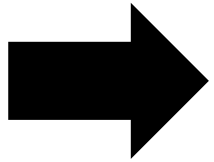
“Carlos”

“Cat”

INPUT **MODEL** **OUTPUT**

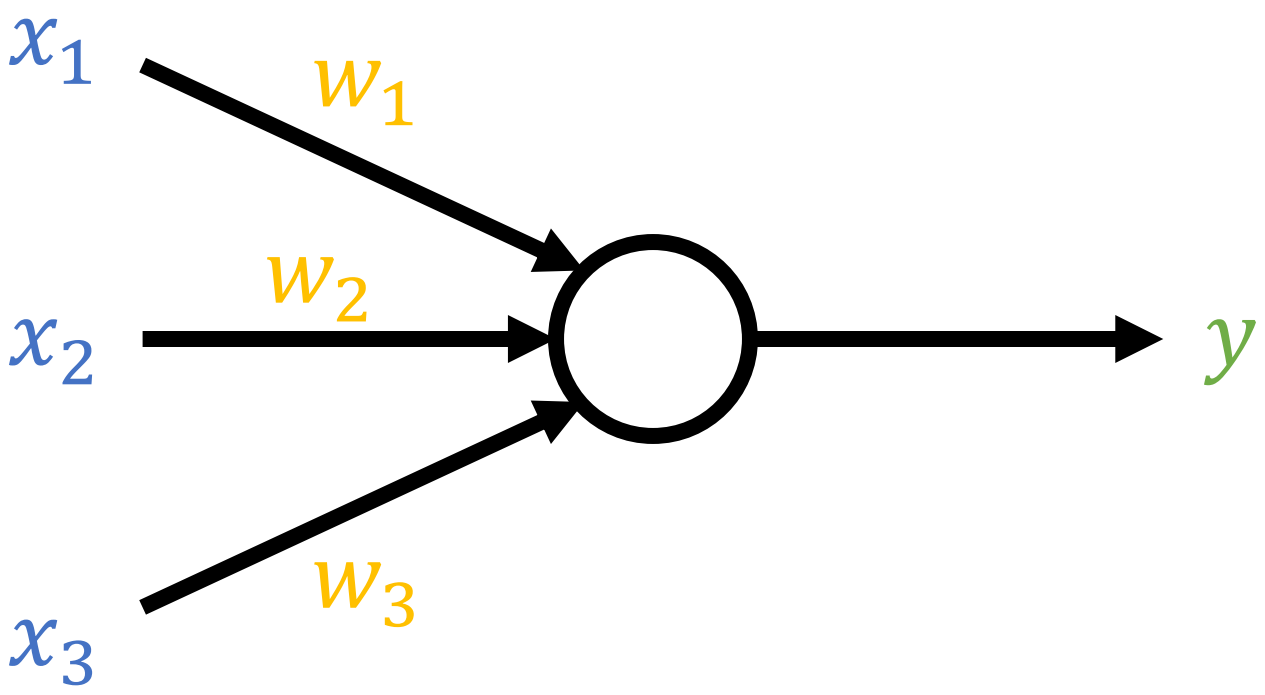




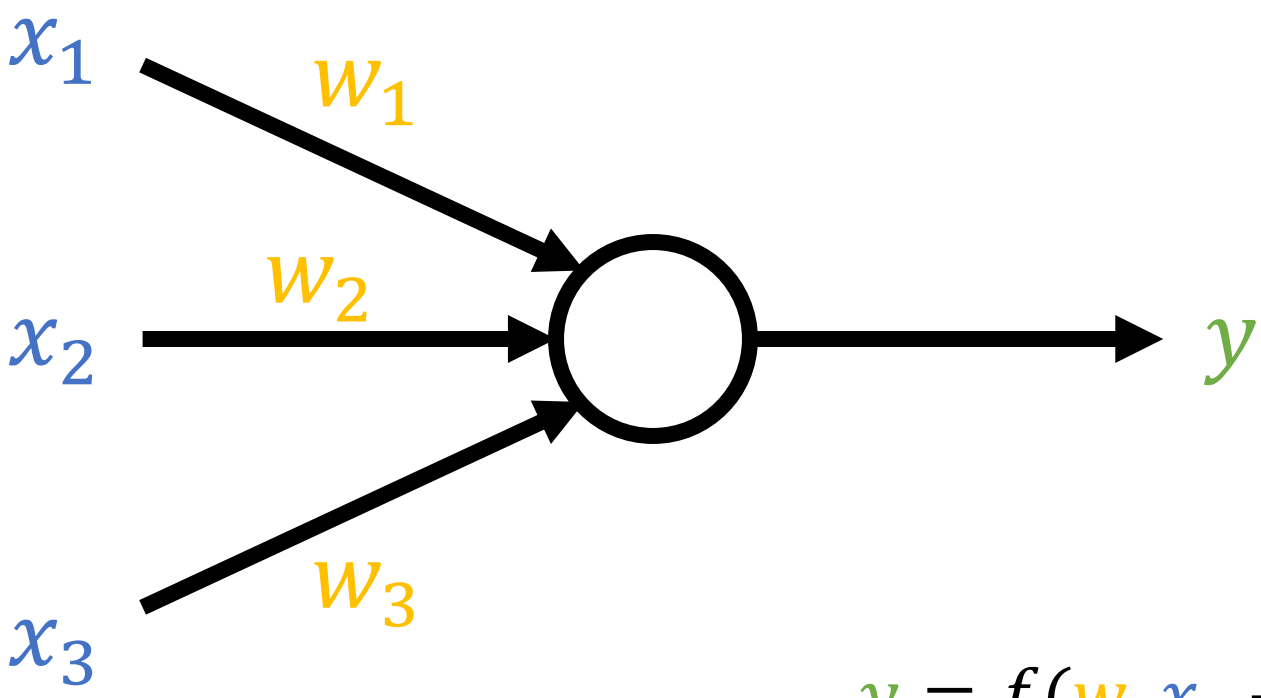


“Cat”

Artificial neural network



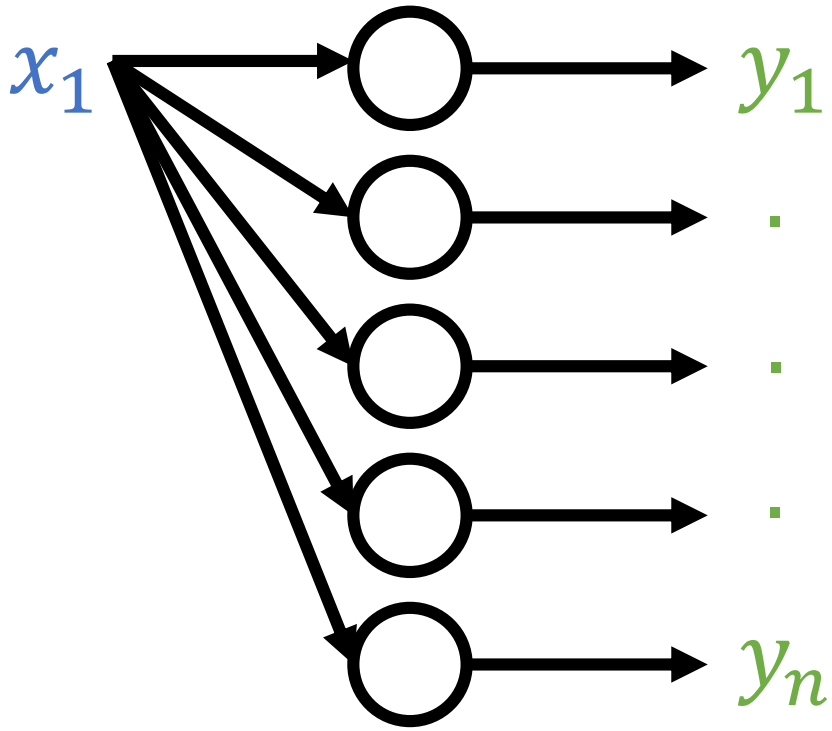
Artificial neural network



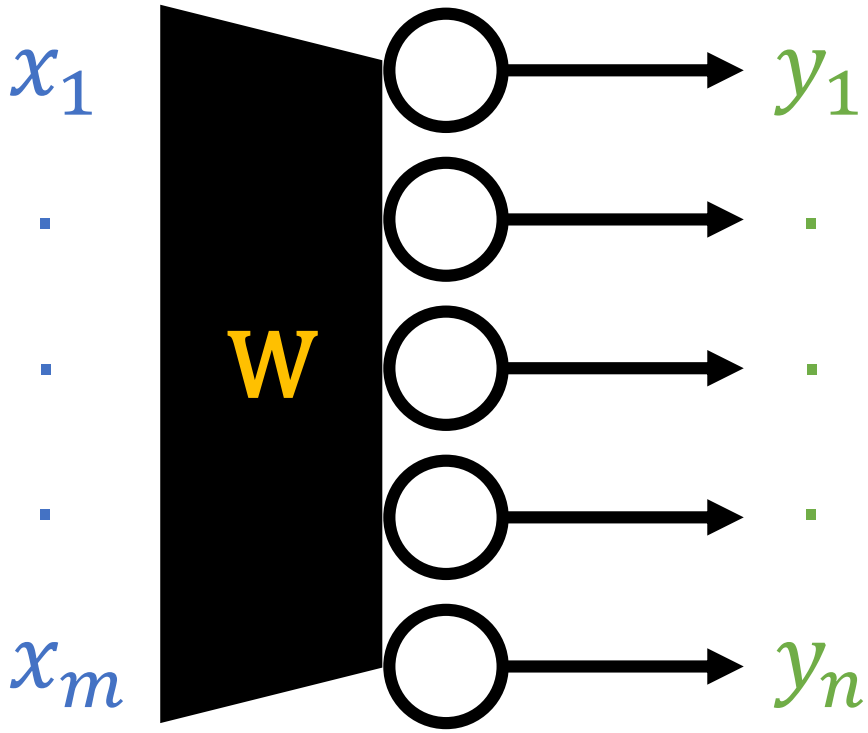
$$y = f(w_1x_1 + w_2x_2 + w_3x_3)$$

$$= f\left(\sum_i w_ix_i\right)$$

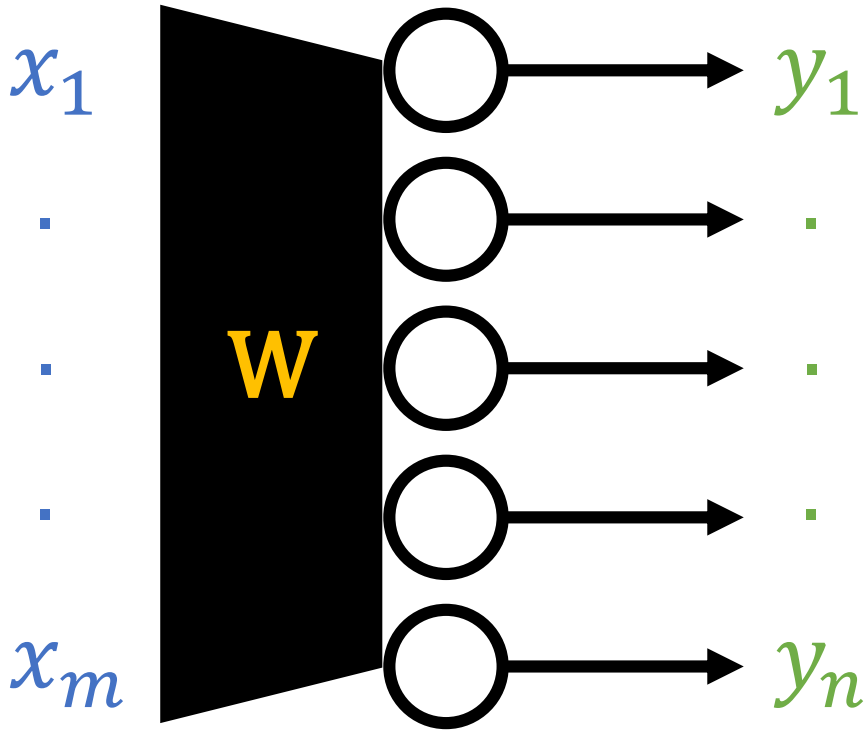
Artificial neural network



Artificial neural network

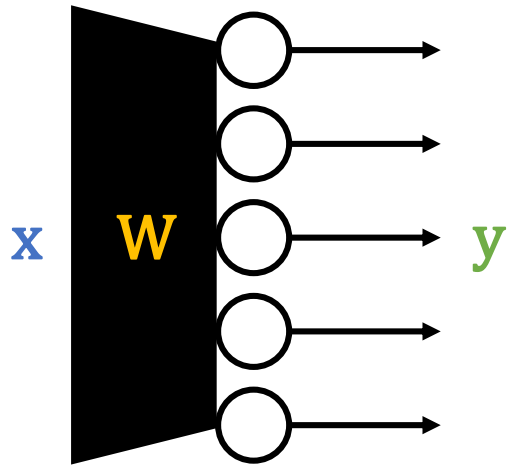


Artificial neural network



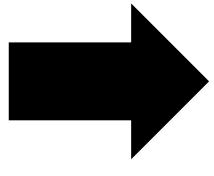
$$\mathbf{y} = f(\mathbf{W}\mathbf{x})$$

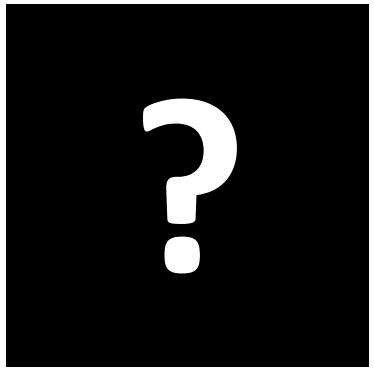
Artificial neural network

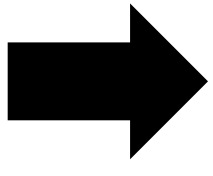


$$\mathbf{y} = f(\mathbf{W}\mathbf{x} + b)$$

Training







“Cat”
“Carlos”
“Cat”

“Cat”
“Cat”
“Cat”

Generalisation

- Test set

Installing TensorFlow

https://github.com/NobodyIsThere/MLWorkshop

Installation instructions in tf_install/ folder. Install Python 3 if necessary.
Linux: included as part of linux.sh.
Windows: Python 3 installer included in tf_install/python/.

Windows:
Run commands in windows.bat.

Linux and Mac:

1. Run commands in first script
2. Run `source ./tf/bin/activate`
3. Run commands in second script

Break?

TensorFlow example (tf_install/test.py)

```
import tensorflow as tf

hello = tf.constant('Hello, world!')

with tf.Session() as sess:
    print(sess.run([hello], feed_dict={}))
```

TensorFlow example (tf_install/test.py)

```
import tensorflow as tf

hello = tf.constant('Hello, world!')

with tf.Session() as sess:
    print(sess.run([hello], feed_dict={}))
```

Create computation graph

List of values to compute

Provide inputs here

Feedforward network example (feedforward.py)

```
x = tf.placeholder(tf.float32, shape=[None, input_length], name="input")
t = tf.placeholder(tf.float32, shape=[None, target_length],name="target")
```

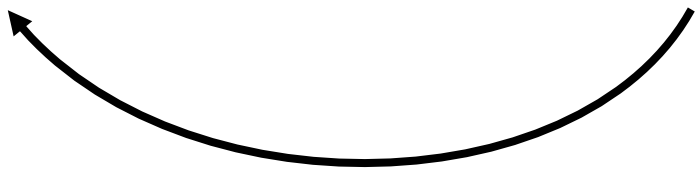
Feed in multiple examples at a time (batches)

```
hidden_layer = tf.nn.relu_layer(x, hidden_weights, hidden_biases)
result = tf.nn.relu_layer(hidden_layer, out_weights, out_biases)

cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(
                                    labels=t, logits=result))
train_op = tf.train.GradientDescentOptimizer(0.01).minimize(cross_entropy)
```

Inputs and outputs

```
x = tf.placeholder(tf.float32, shape=[None, input_length], name="input")
.
.
.
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
.
.
.
with tf.Session() as sess:
    acc, loss = sess.run([accuracy, loss_var], feed_dict={x: value_of_x})
```

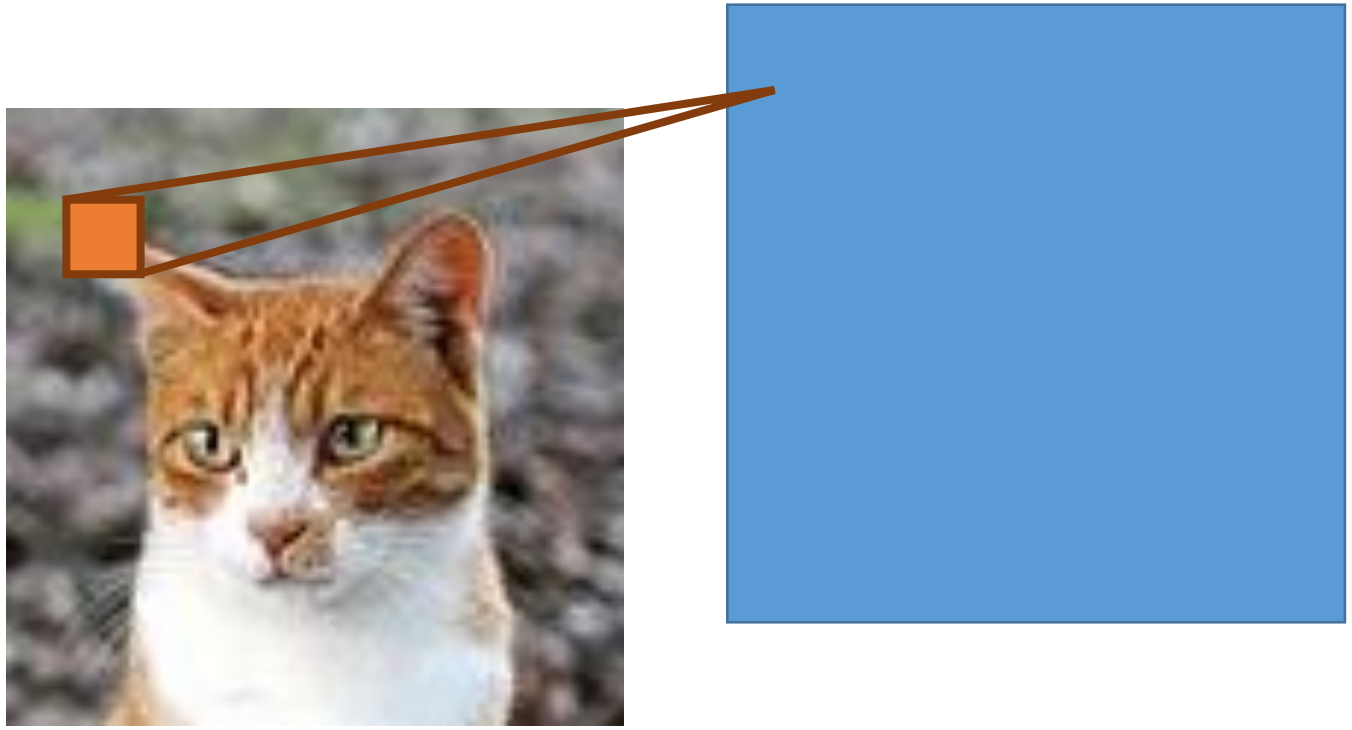


Convolutional neural networks

- Spatial dependence
- Weight sharing

Convolutional neural networks

- Spatial dependence
- Weight sharing

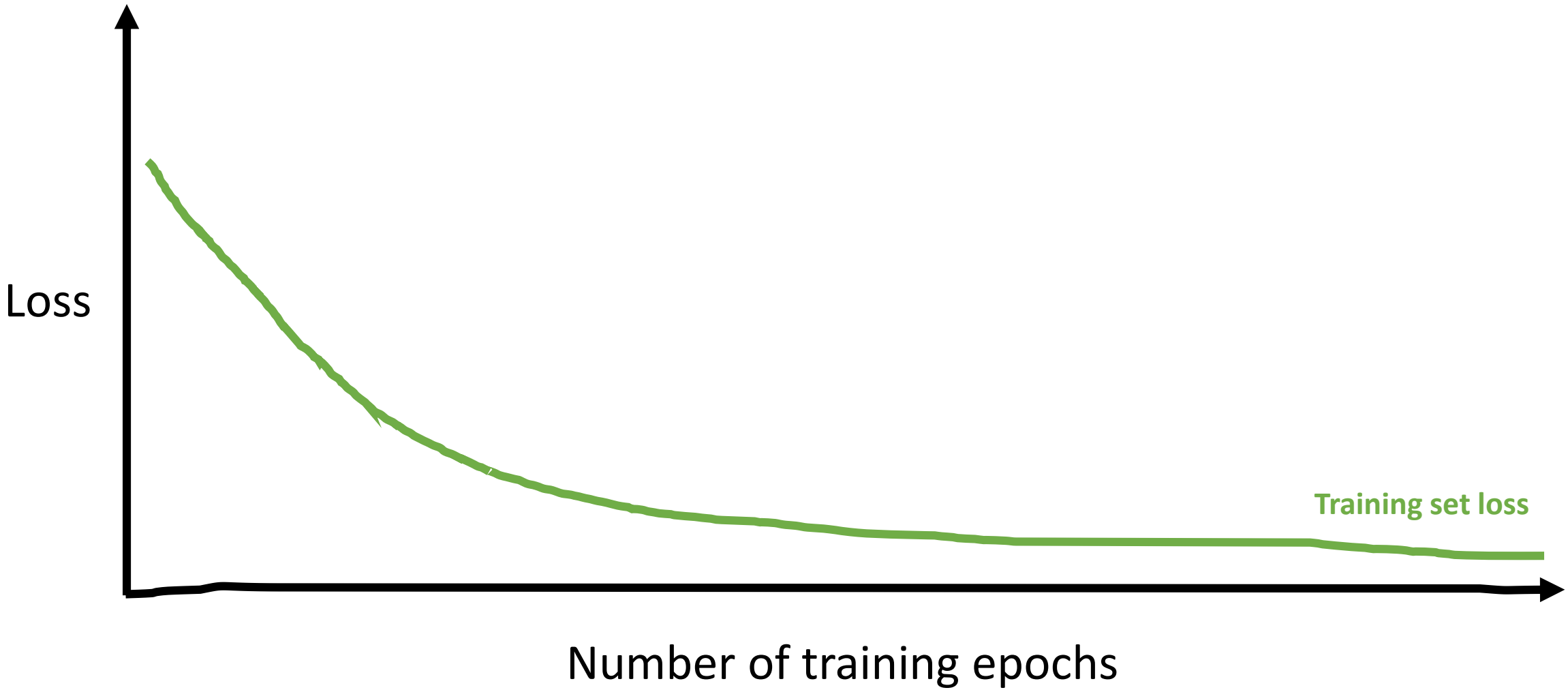


Convolutional neural networks

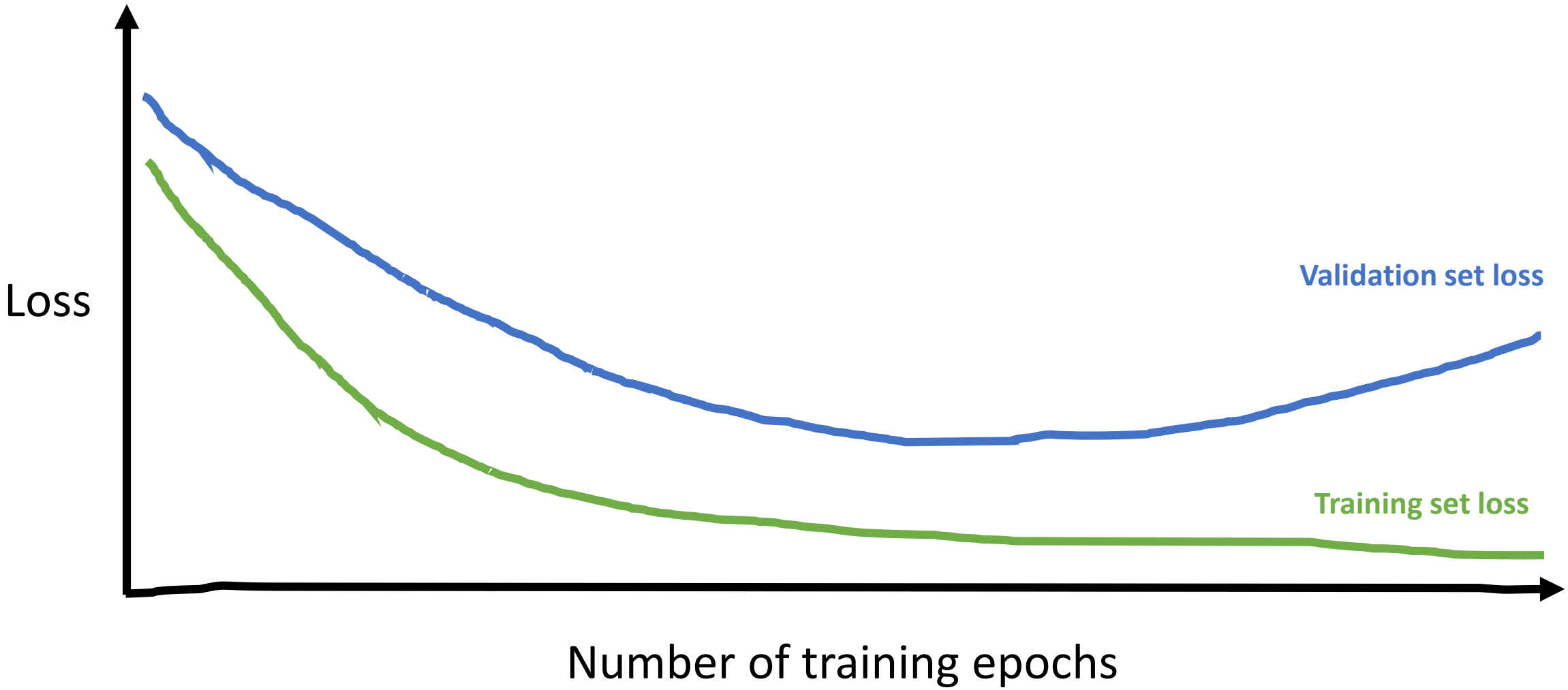
```
conv = tf.nn.relu(tf.nn.conv2d(x, W, strides=[1, 1, 1, 1],
                               padding='SAME') + b)

pool_layer = tf.nn.max_pool(conv, ksize=[1, 2, 2, 1],
                            strides=[1, 2, 2, 1],
                            padding='SAME')
```

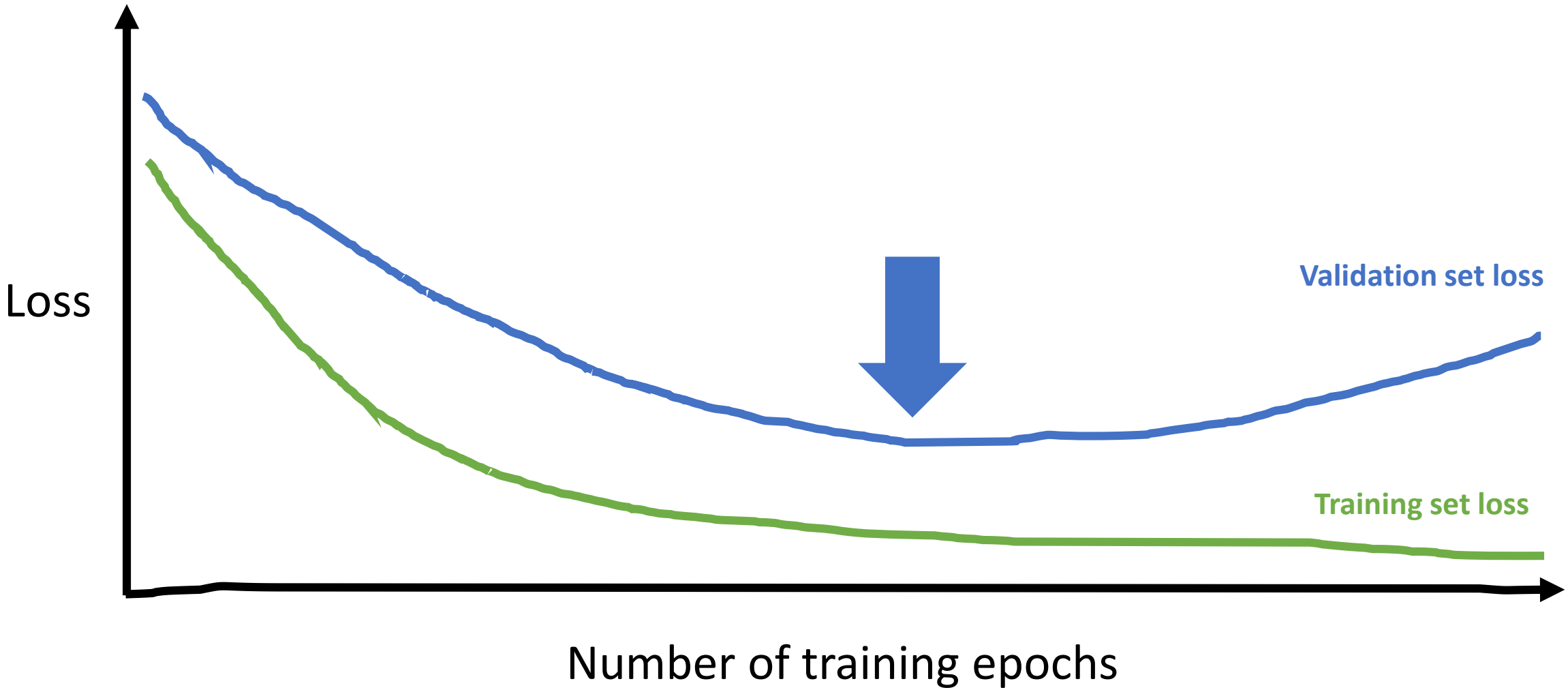
Regularisation



Regularisation



Regularisation



Regularisation

- Early stopping
- Add terms to cost function
- Dropout

Further reading

- Neural networks: neuralnetworksanddeeplearning.com
- Deep learning book: *Deep Learning* – Goodfellow, Bengio & Courville
- Convolutional nets: Stanford course at cs231n.github.io – also covers data preprocessing

References

Image credit

Cat pictures: various (commons.wikimedia.org/wiki/File:Cat_poster_1.jpg)

I AM PLAYR: twitter.com/iamplayr

Lyroke: twitter.com/lyroke

Race Team Manager: facebook.com/rtmgame

AI Factory *Spades:* aifactory.co.uk/AIF_Games_Spades.htm