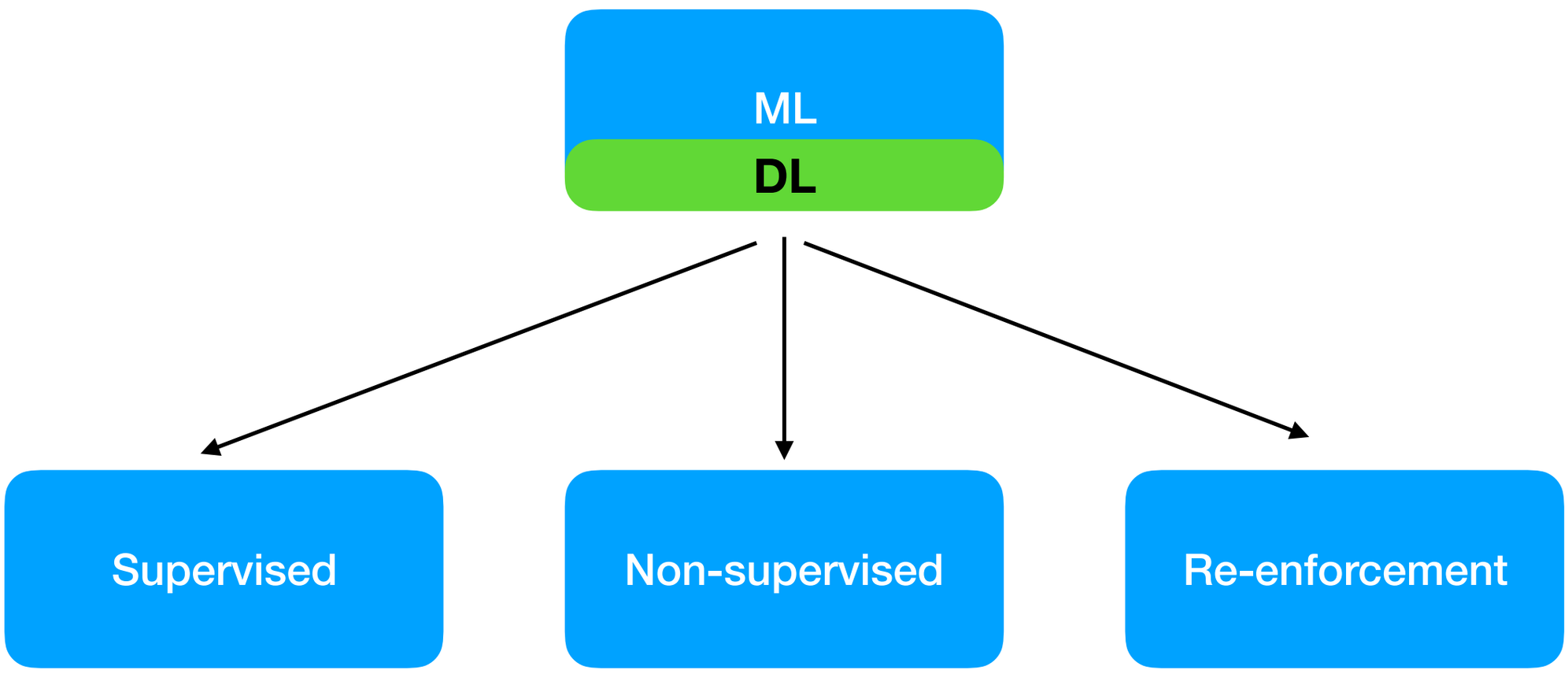
Deep Learning

A friendly introduction

An Truong
Lincoln, 2019

Deep Learning vs Machine Learning



- House/car's price prediction
- Fraud detection etc…

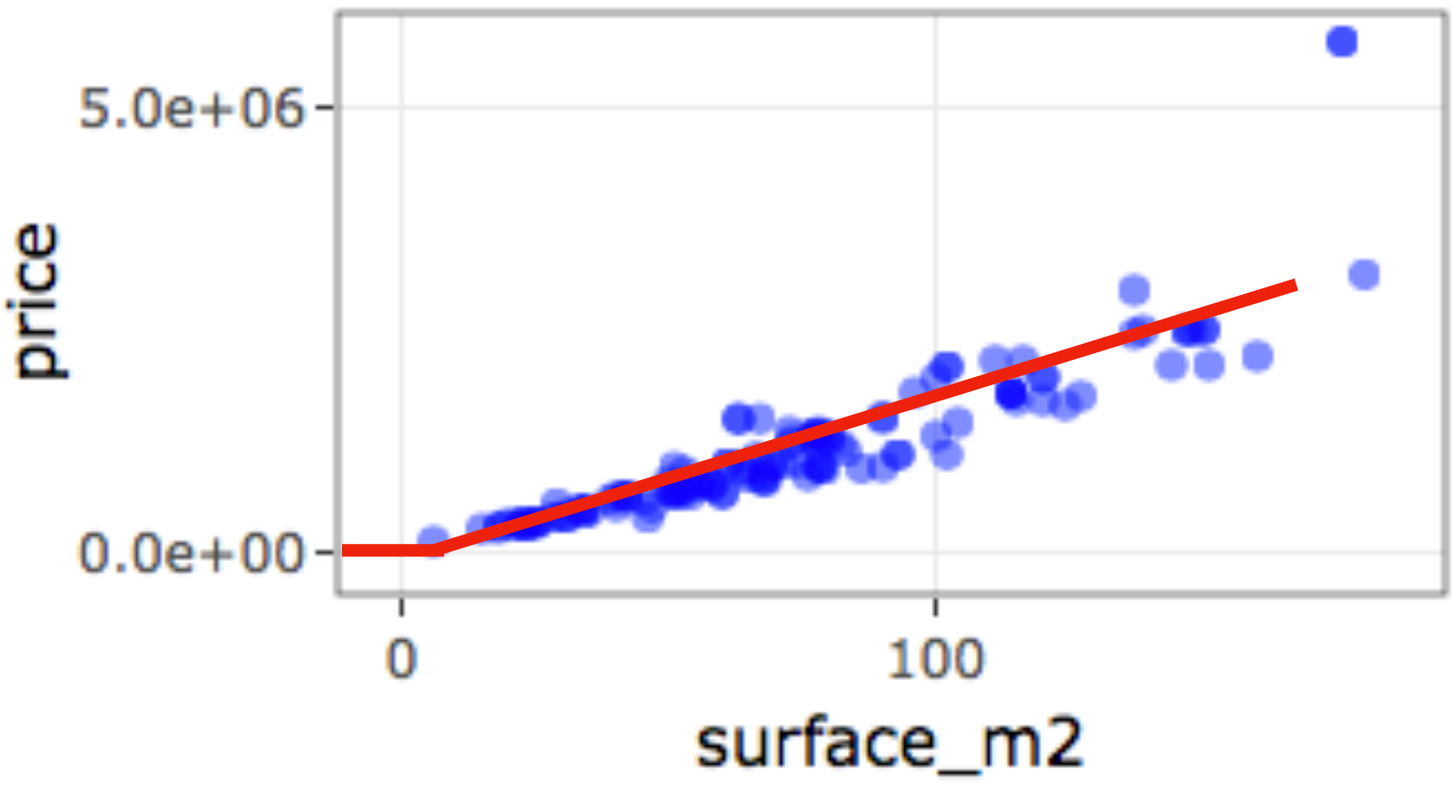
- Market segmentation
- Client profiling etc…

- Seft-driving car
- Robotic, etc…

next

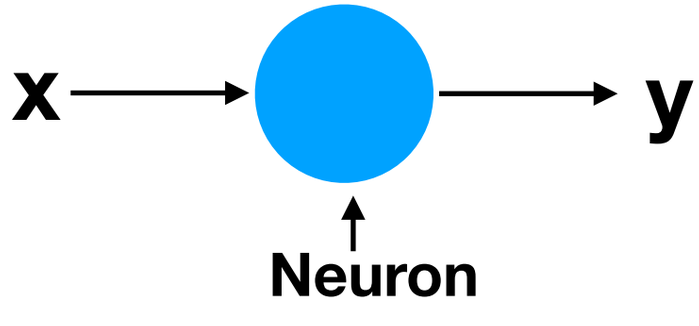
Simplest NN

credit to A. Ng

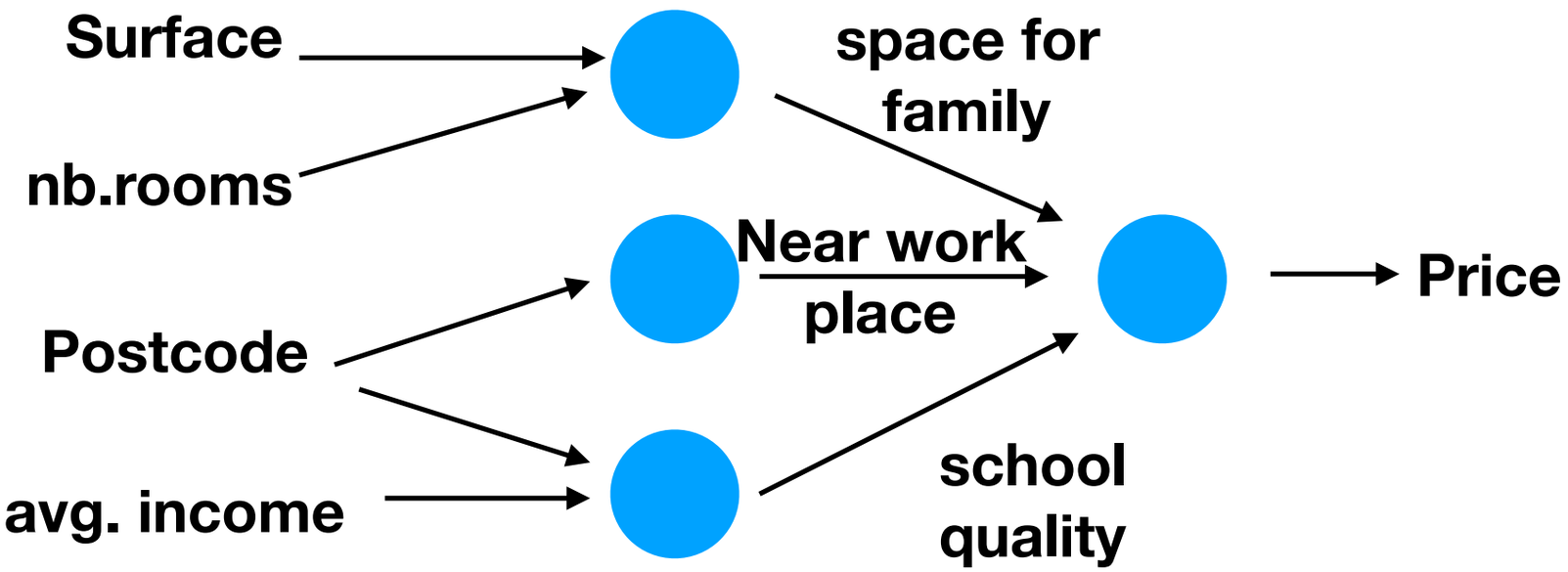


ReLu

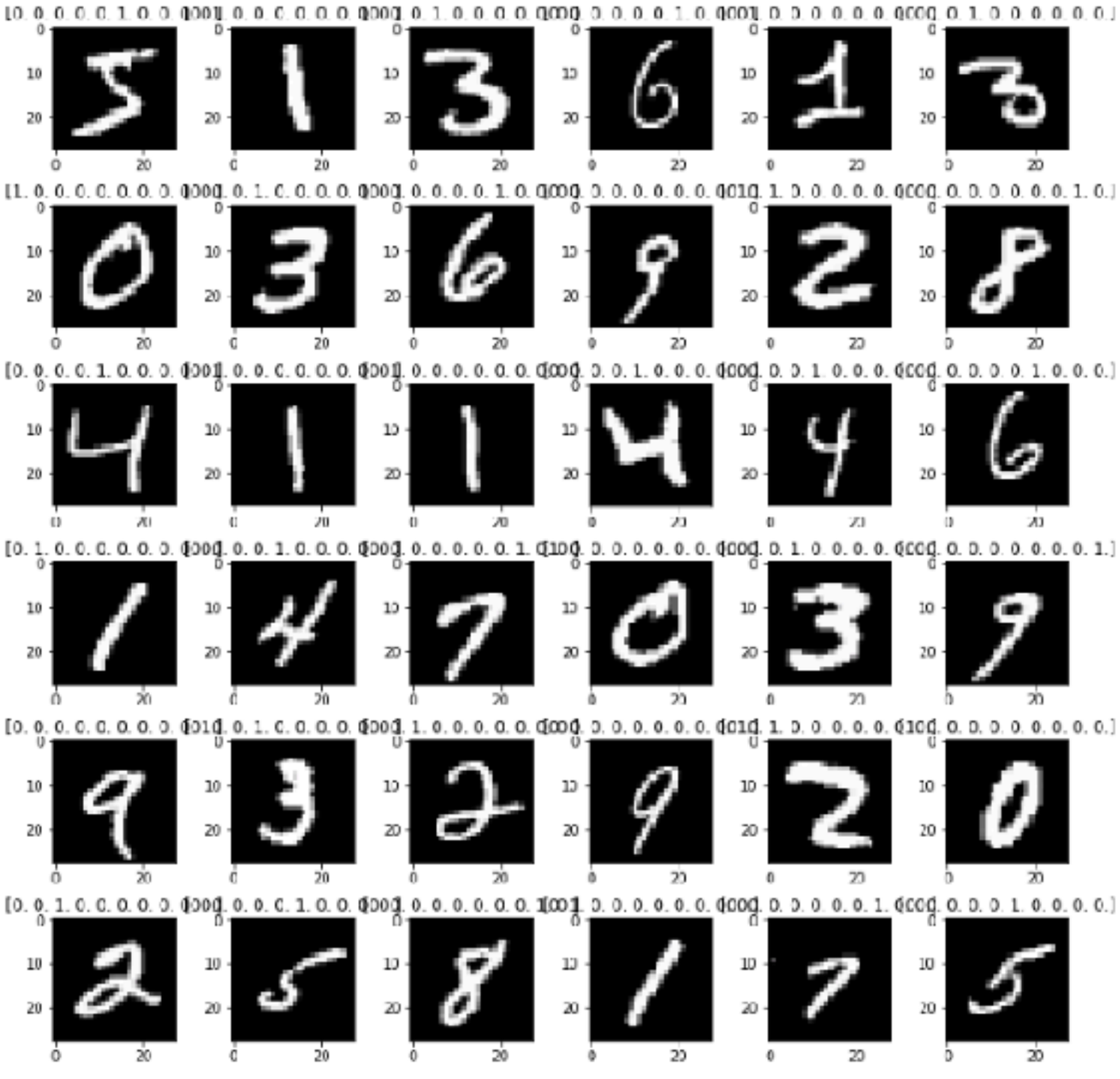
y = relu(x*w + b)

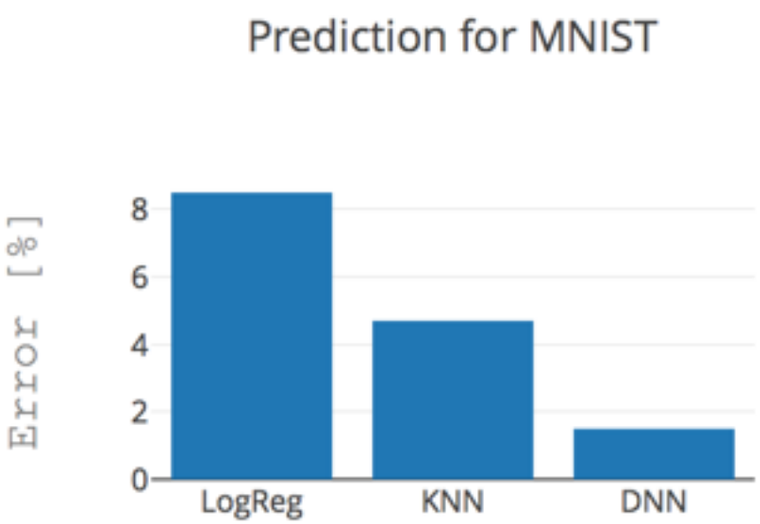


Multilayer NN



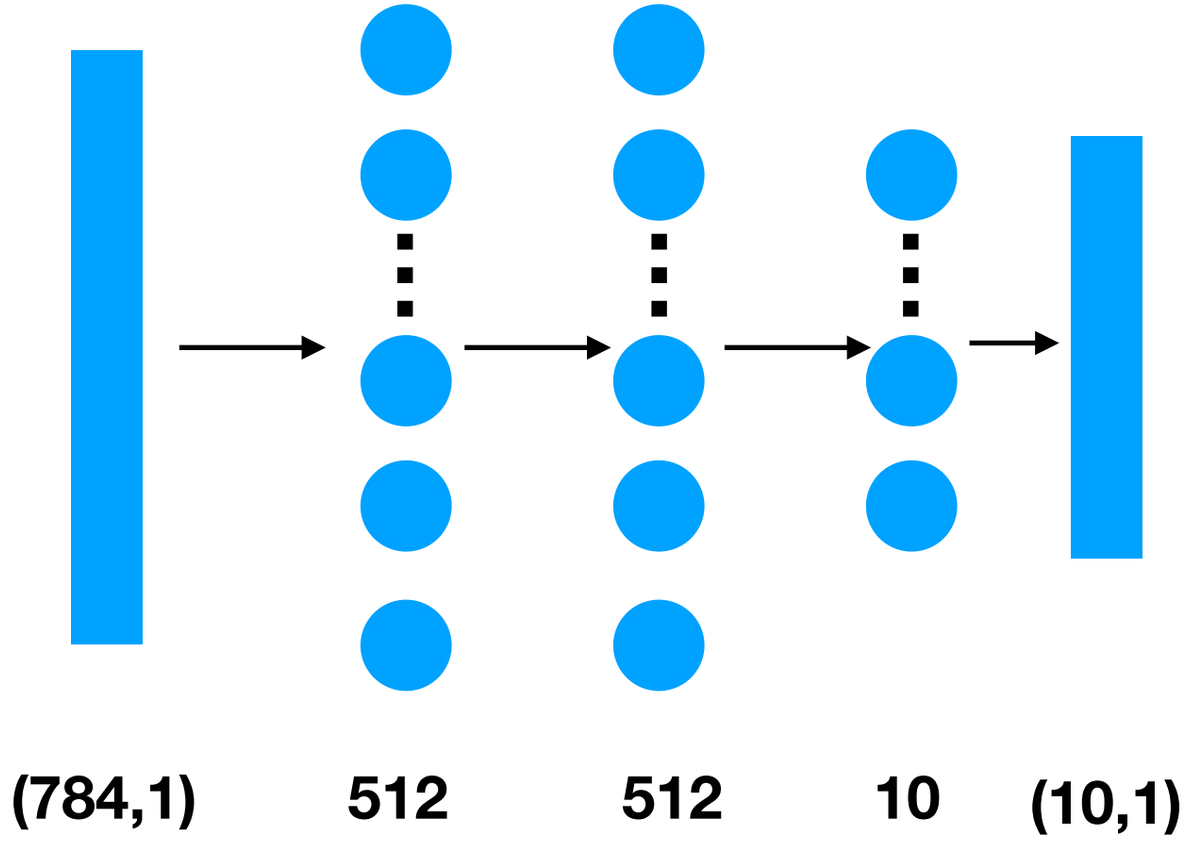
Hand on: MNIST





NN-Model

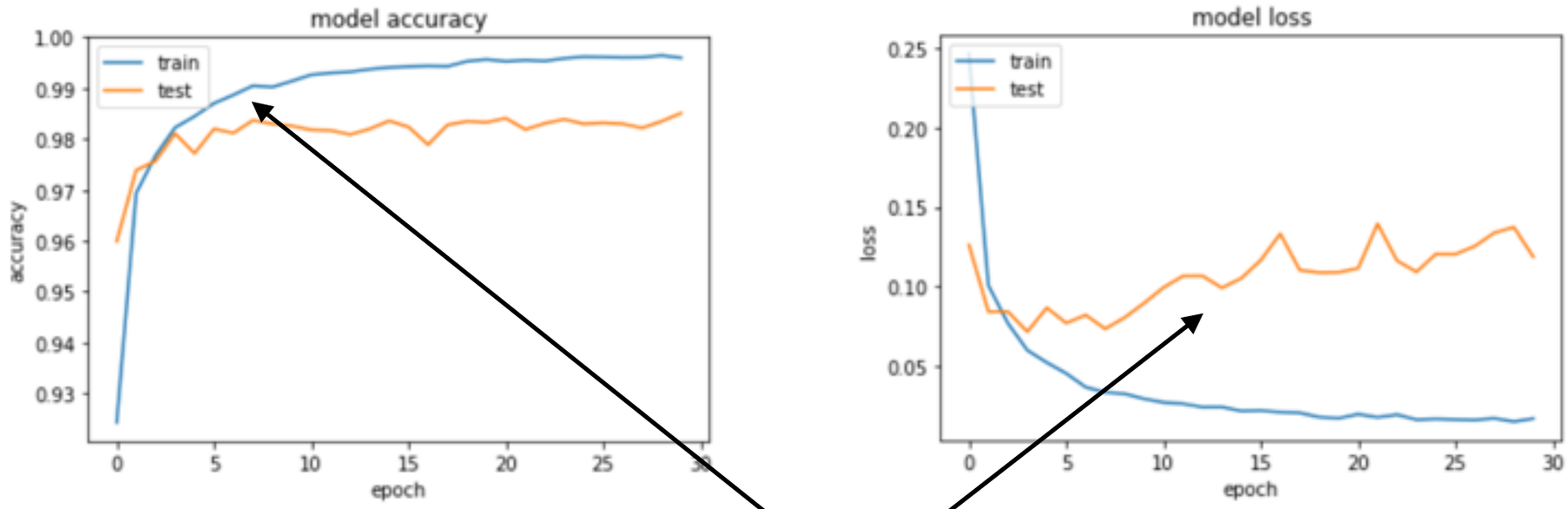
```
model = Sequential()
model.add(Dense(512, activation='relu', input_shape=(784,)))

model.add(Dense(512, activation='relu'))

model.add(Dense(num_classes, activation='softmax'))
```



Training

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/30
60000/60000 [==============================] - 17s 277us/step - loss: 0.2460 - acc: 0.9243 - val_loss: 0.1263 - val_
acc: 0.9599
Epoch 2/30
60000/60000 [==============================] - 18s 304us/step - loss: 0.1009 - acc: 0.9694 - val_loss: 0.0844 - val_
acc: 0.9739
Epoch 3/30
60000/60000 [==============================] - 15s 254us/step - loss: 0.0772 - acc: 0.9768 - val_loss: 0.0846 - val_
acc: 0.9757
Epoch 4/30
60000/60000 [==============================] - 15s 248us/step - loss: 0.0603 - acc: 0.9823 - val_loss: 0.0719 - val_
acc: 0.9811
Epoch 5/30
60000/60000 [==============================] - 15s 249us/step - loss: 0.0526 - acc: 0.9845 - val_loss: 0.0869 - val_
acc: 0.9772
Epoch 6/30
60000/60000 [==============================] - 16s 267us/step - loss: 0.0457 - acc: 0.9870 - val_loss: 0.0773 - val_
acc: 0.9820
Epoch 7/30
60000/60000 [==============================] - 16s 261us/step - loss: 0.0371 - acc: 0.9887 - val_loss: 0.0824 - val_
acc: 0.9812
```

Overfitting



Overfitting

deploy