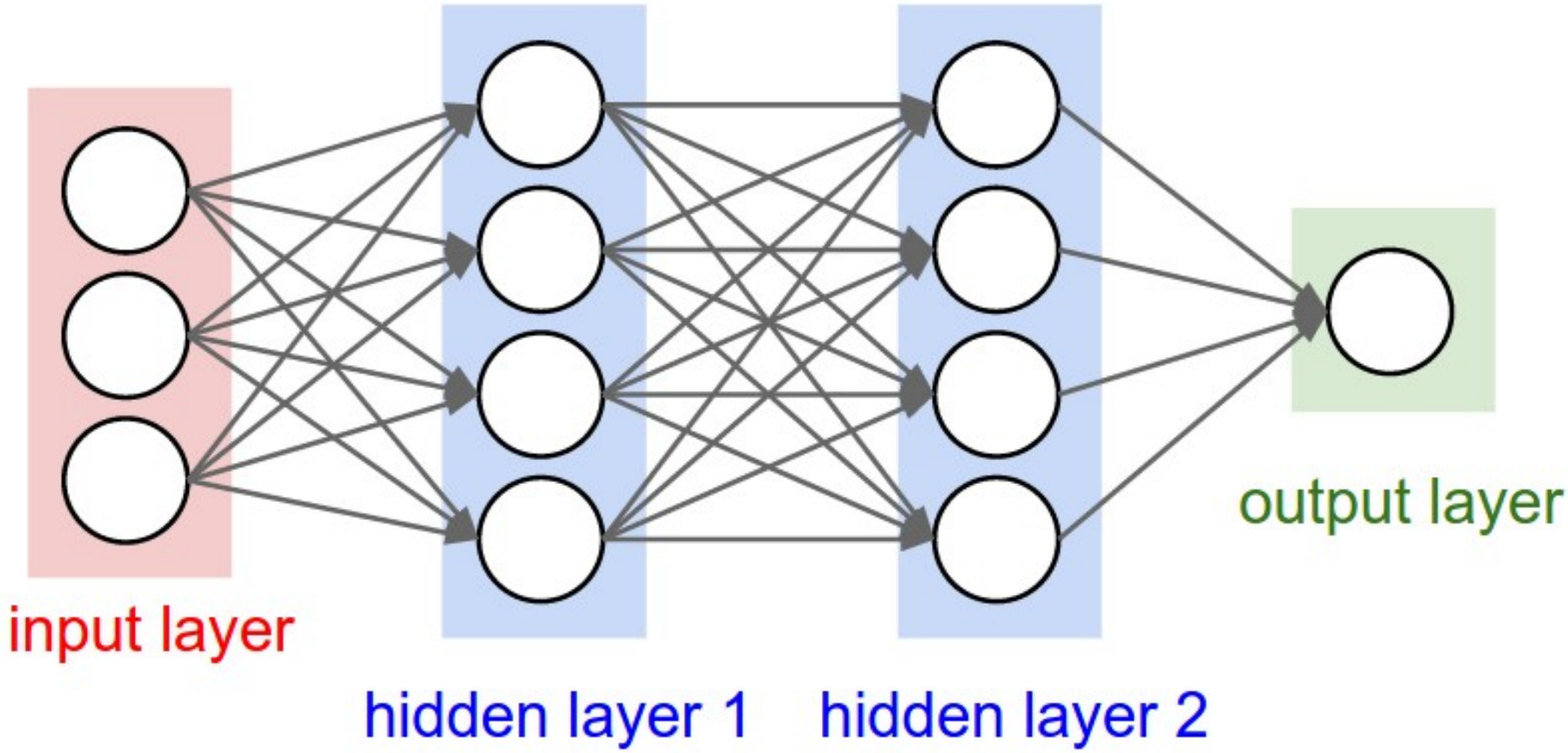
Neural Networks

Working with Keras

Term : Dense Layer

- Also known as fully connected layer
- All the nodes in this layer are connected to all other nodes in the previous layer



Term : Optimizers

- Optimizers are the algorithms which reduce the loss
- eg. Gradient Descent
- There are many others
  - Adam Optimizer
  - Momentum
  - Nesterov
  - SGD (Stochastic Gradient Descent)

- Details about each of them is here

Term : Loss

- Represent the loss function which we use for our problem
- Can be of different type for different problems
  - For **regression** - mean_squared_error
  - For **classification** - ??
- List of loss function is available here

Keras : Steps

1. Specify Architecture
    1. Layers
    2. Activation Functions
2. Compile
    1. Specify loss and optimizers
3. Fit
    1. Train the data and calculate loss
    2. Optimize
4. Predict

Keras : Architecture

```
1  # Import necessary modules
2  import keras
3  from keras.layers import Dense
4  from keras.models import Sequential
5
6  # Save the number of columns in predictors: n_cols
7  n_cols = predictors.shape[1]
8
9  # Set up the model: model
10 model = Sequential()
11
12 # Add the first layer
13 model.add(Dense(50, activation='relu', input_shape=(n_cols,)))
14
15 # Add the second layer
16 model.add(Dense(32,activation='relu'))
17
18 # Add the output layer
19 model.add(Dense(1))
20
```

Nodes

Input shape

Layer 1

Activation

Layer 2

Output Layer

Keras : Compile

Optimizer

Loss Function

```
1 # Compile the model
2 model.compile(optimizer='adam',loss='mean_squared_error')
3
4 # Verify that model contains information from compiling
5 print("Loss function: " + model.loss)
```

```
Loss function: mean_squared_error
```

Keras : Fit the model

```
# Fit the model
model.fit(predictors,target)
```

Training the model

```
Epoch 1/10
534/534 [==============================] - 0s - loss: 187.4750
Epoch 2/10
534/534 [==============================] - 0s - loss: 35.5615
Epoch 3/10
534/534 [==============================] - 0s - loss: 27.6003
Epoch 4/10
534/534 [==============================] - 0s - loss: 23.8839
Epoch 5/10
534/534 [==============================] - 0s - loss: 22.9858
Epoch 6/10
534/534 [==============================] - 0s - loss: 22.5383
Epoch 7/10
534/534 [==============================] - 0s - loss: 22.1515
Epoch 8/10
```

Number of iterations

Loss

Keras : Classifcation

Things have changed now

Points to think about

- How many nodes in the output?
- How do we know which class is which?
- Do we need to prepare the data?
- Loss function ? - MSE will not work now.

Keras : Classifcation

```
# Specify, compile, and fit the model
model = Sequential()
model.add(Dense(32, activation='relu', input_shape = (n_cols,)))
model.add(Dense(2, activation='softmax'))
model.compile(optimizer='sgd',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.fit(predictors, target)

# Calculate predictions: predictions
predictions = model.predict(pred_data)

# Calculate predicted probability of survival: predicted_prob_true
predicted_prob_true = predictions[:,1]

# print predicted_prob_true
print(predicted_prob_true)
```

Softmax

Loss function changed

Predictions