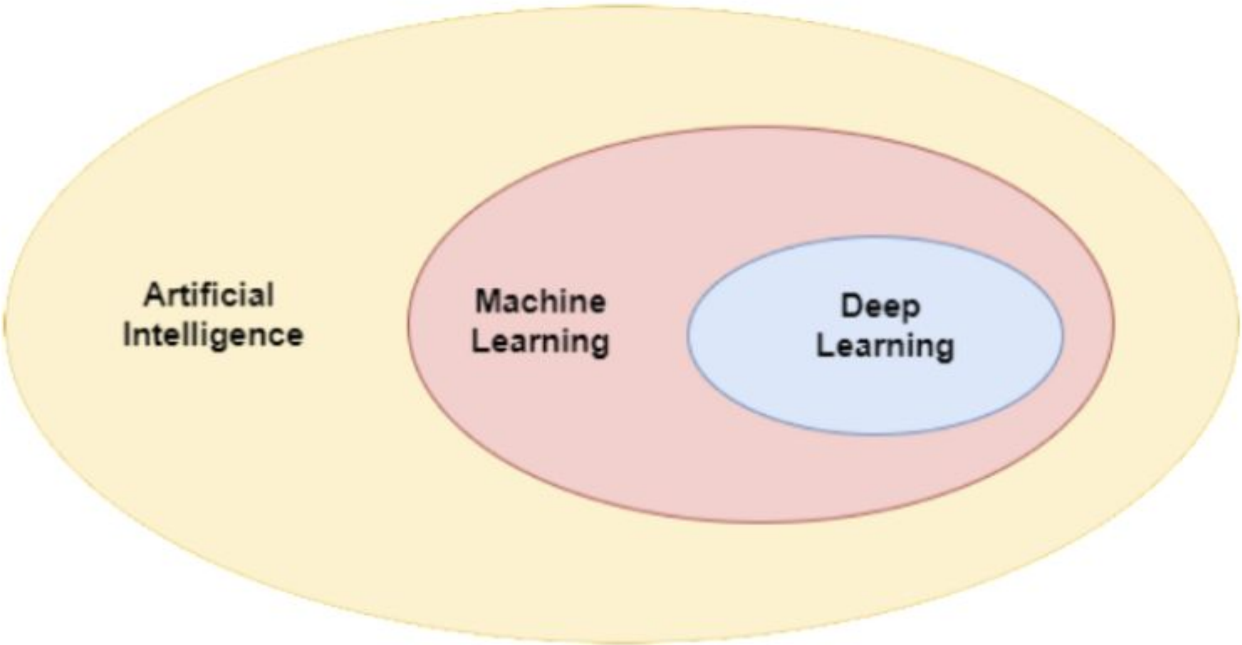
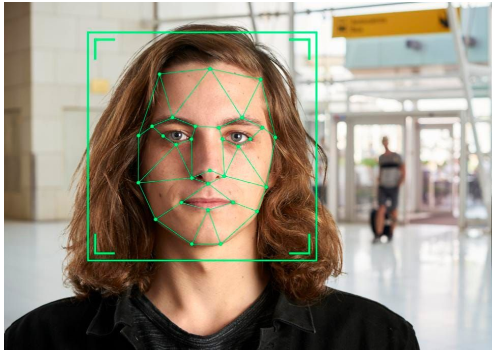
Deep Learning Workshop

BUDSA – Ryan and Wei



https://towardsdatascience.com/understanding-the-difference-between-ai-ml-and-dl-cceb63252a6c

Real Life Applications

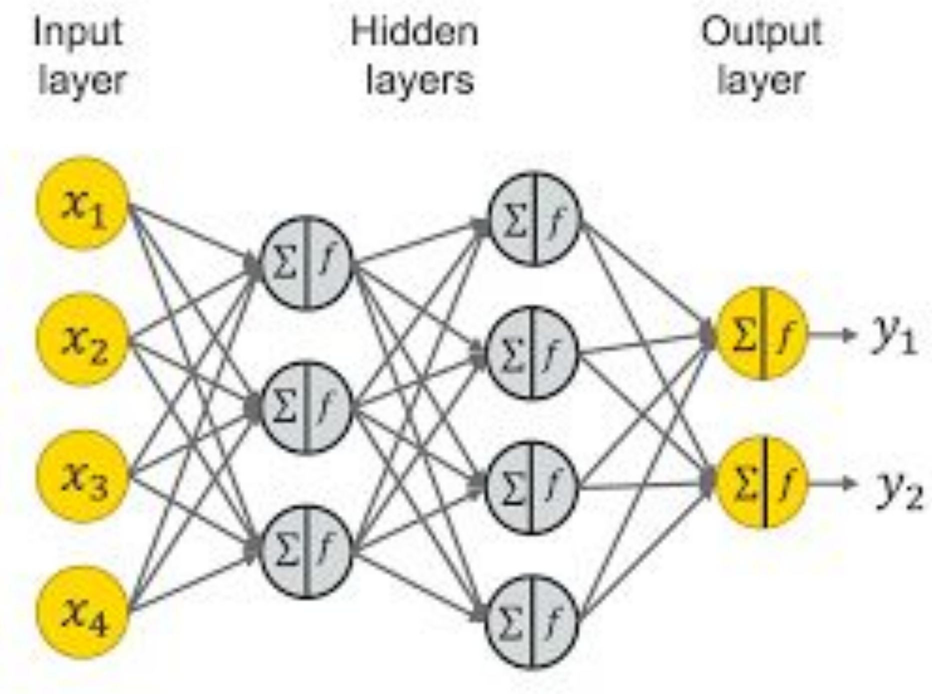


Loss function

Activation function, softmax

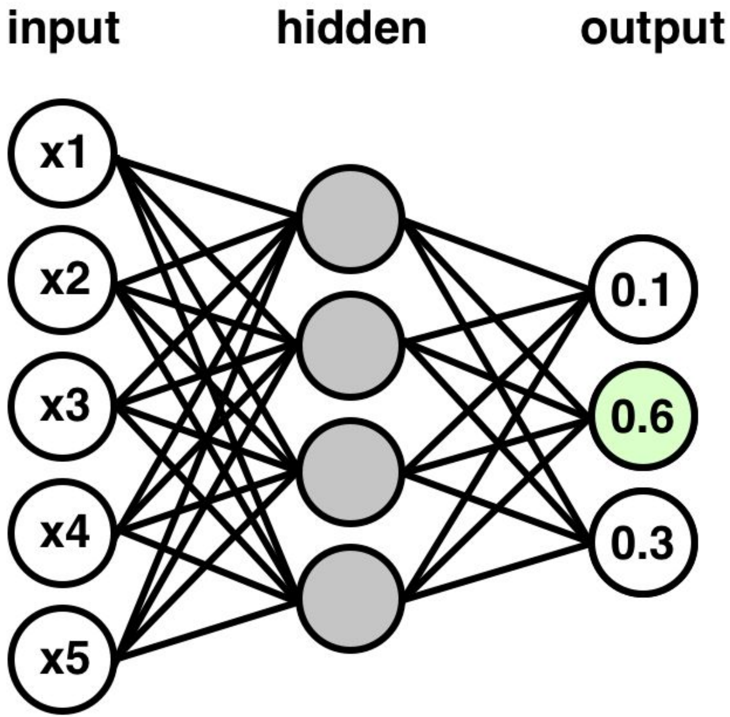
Gradient descent

Inputs & transformations - What is happening to the data?



- Input features as $\{x_1, x_2, x_3, x_4\}$
- Goes through linear transformations – **wX + b = a**
- Then goes through nonlinear transformations (ReLU, Tanh, softmax, etc.) – **f(a)**
- Linear transformation again – **$w_2f(a) + b_2 = a_2$**
- Now nonlinear – **$g(a_2)$** and outputs

Output



Now you know how a neural network works, let's create and train one

mxnet

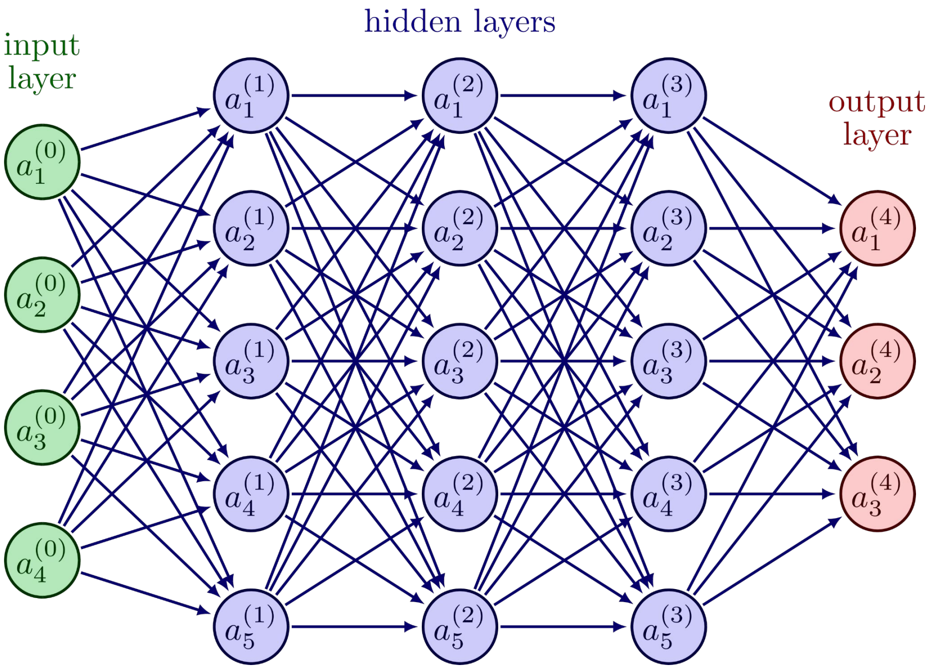
TensorFlow

PyTorch

Keras

Caffe

Creating a Simple Neural Network Architecture



Tensorflow code here

```
# Create features
X = np.array([-7.0, -4.0, -1.0, 2.0, 5.0, 8.0, 11.0, 14.0])

# Create labels
y = np.array([3.0, 6.0, 9.0, 12.0, 15.0, 18.0, 21.0, 24.0])
```

```
# Split data into train and test sets
X_train = X[:40] # 80 % of data
y_train = y[:40] # 80 % of data

X_test = X[40:] # last 10 examples -> (20% of data)
y_test = y[40:]

len(X_train), len(X_test)
```

```
(8, 0)
```

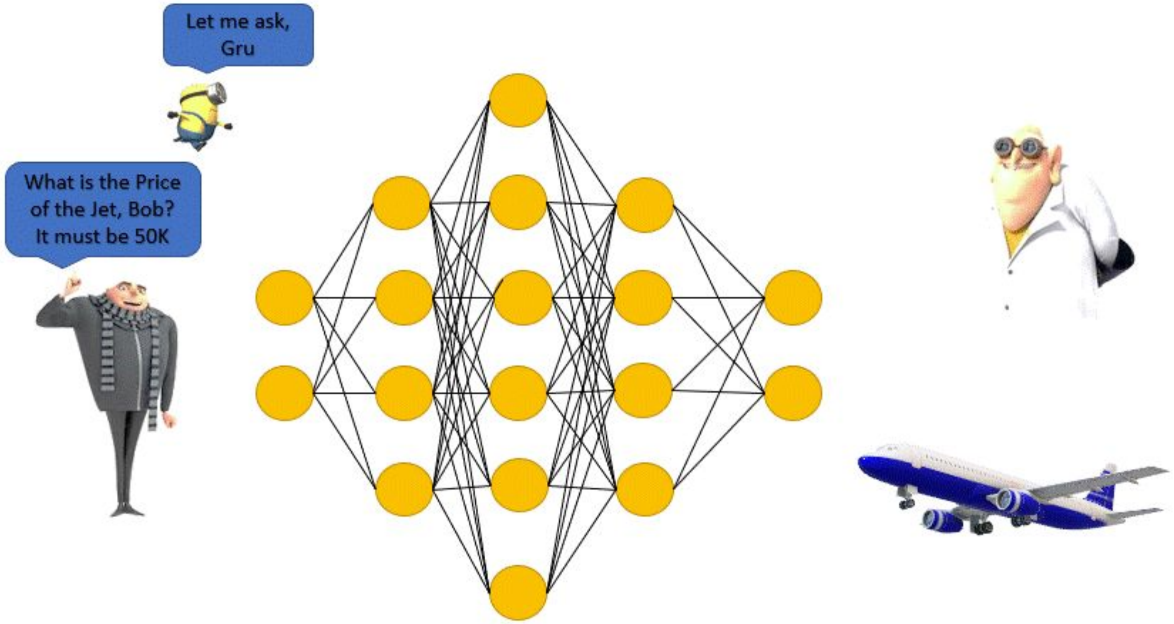
```
# Create a model (same as above)
model = tf.keras.Sequential([
    tf.keras.layers.Dense(100, input_shape=[1]),
    tf.keras.layers.Dense(100),
    tf.keras.layers.Dense(100),
    tf.keras.layers.Dense(10),
    tf.keras.layers.Dense(1)
])

# Compile model (same as above)
model.compile(loss=tf.keras.losses.mae,
              optimizer=tf.keras.optimizers.Adam(),
              metrics=['mae'])

# fit the model and save the history (we can plot this)
history = model.fit(X_train, y_train, epochs=300)
```

$$\ell_{\mathrm{MSE}}(\hat{\mathbf{y}}, \mathbf{y}) = ||\hat{\mathbf{y}} - \mathbf{y}||^2$$

Training – Backpropagation



$$\ell_{\text{MSE}}(\hat{\mathbf{y}}, \mathbf{y}) = ||\hat{\mathbf{y}} - \mathbf{y}||^2$$

$$\frac{\partial \ell}{\partial W^{(2)}} = \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W^{(2)}} = \frac{\partial \ell}{\partial \hat{y}} a$$