Processing Biological Data

Prof. Dr. Volkhard Helms — Chair for Computational Biology Saarland University
Dr. Pratiti Bhadra — Tutor
Summer Semester 2020

Assignment Sheet 4
Deep Learning
Due: June XX, 2020 10:15 am

1. **Exercise 4.1: Basics of Deep learning methods** [50 points]

   (a) Briefly explain what is meant by overfitting. How to avoid overfitting in deep neural networks? (5 point) **Answer:** Overfitting is a modelling error that occurs when a function is too closely fit to a limited set of data points. A model that learns the training dataset too well, performing well on the training dataset but does not perform well on a hold out sample. There are two ways to approach an overfit model: (1) Reduce overfitting by training the network on more examples. (2) Reduce overfitting by changing the complexity of the network (controlling hyper-parameter,weight regularization,dropout,early stopping, ensemble model, noise etc.).

   (b) What is dropout in neural network? Can it be applied at visible layer (or input layer) of neural network? Which of the following statement is true for dropout? (5 points)

   i. Dropout gives a way to approximate by combining many different architectures
   ii. Dropout can help preventing overfitting
   iii. Dropout prevent hidden unit from coadapting

   Answer: Dropout refer to dropping out units (neuron) randomly. It prevent network from over-fitting.Yes, it can apply in both visible and hidden layers. All statements are true.

   (c) What is activation function of neural network? What is the purpose of the activation function in neural network? What are advantages of ReLu function over Sigmoid function? (5 points) **Answer:** An activation function determines the output behavior of each node, or neuron in an artificial neural network. The purpose of the activation function is to introduce non-linearity into the output of a neuron. ReLu solve the vanishing gradient problem, more computationally efficient (faster) than sigmoid.

   (d) Cross validation: Carry out leave-one-out cross-validation (LOOCV) in a simple classification problem. Consider the following dataset with one real-valued input x (numbers on the line in the figure) and one binary output y (negative and positive sign). We are going to use k-NN with Euclidean distance to predict $\hat{y}$ for x.

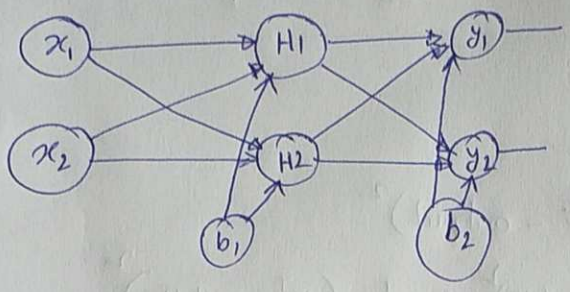
   What is the LOOCV error of 1-NN on this dataset? Give your answer as the total number of misclassifications. (5 points) **Answer:** 6 (0.7, 1.0, 2.5, 3.2, 3.5, 4.1)

   (e) Determine thevalues at hidden and output layers if x1 = 0.05, x2 = 0.10, w1 = 0.15, w2 = 0.20, w3 = 0.25, w4 =0.30, w5 = 0.40, w6 = 0.45, w7 = 0.50, w8 = 0.55, b1= 0.35 and

b2=0.60. Activation function is sigmoid (logistic function $f(x) = \frac{L}{1+e^{-x}}$). Determine the total error (the mean squared error) if the target (or actual) outputs are $y1^T = 0.01$ and $y2^T = 0.99$. What is the updated weight of w5 after first back-propagation on this example? learning rate is 0.5. (10+5+10 = 25 points)

Answer: see next page

(Ex4.4) (e)



$x_1 = 0.05$, $x_2 = 0.10$, $b_1 = 0.35$ $b_2 = 0.60$

$w_1 = 0.15$, $w_2 = 0.20$, $w_3 = 0.25$ $w_4 = 0.30$

$w_5 = 0.40$, $w_6 = 0.45$, $w_7 = 0.50$, $w_8 = 0.55$

hidden and output node function

example H1

$$H1 = x_1 \times w_1 + x_2 \times w_2 + b_1$$

Activation function is sigmoid = $\frac{1}{1+e^{-x}}$

output of from H1

$$= \frac{1}{1+e^{-H_1}}$$

(Forward Popagation)

$$H1 = 0.05 \times 0.15 + 0.10 \times 0.20 + 0.35$$

$$= 0.377$$

$$outH1 = \frac{1}{1+e^{-0.377}} = 0.593269992 \approx \boxed{0.5932}$$

Samewday $outH2 = 0.596884378 \approx \boxed{0.5968}$

$$y1 = outH_1 * w_5 + outH_2 * w_6 + b_2 = 1.1059$$

$$outy1 = \frac{1}{1+e^{-y1}} = \frac{1}{1+e^{-1.1059}} \approx \boxed{0.7513}$$

Samway $outy2 \approx \boxed{0.7729}$

Target value $y_1^T = 0.01$ $y_2^T = 0.99$

Calculate the error

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

$$= \underbrace{\frac{1}{2}(y_1^T - out\,y_1)^2}_{E_1} + \underbrace{\frac{1}{2}(y_2^T - out\,y_2)^2}_{E_2}$$

$$= 0.298371109 \approx \boxed{0.29837}$$

Backpropagation. update w_5

$$\text{Calculate error at } w_5 = \frac{\partial E_{total}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out\,y_1} * \frac{\partial out\,y_1}{\partial y_1} * \frac{\partial y_1}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial out\,y_1} = \frac{d\left(\frac{1}{2}(y_1^T - out\,y_1)^2 + \frac{1}{2}(y_2^T - out\,y_2)^2\right)}{d\,out\,y_1}$$

$$= 2 * \frac{1}{2}(y_1^T - out\,y_1)^{2-1} * -1 + 0$$

$$= out\,y_1 + y_1^T = 0.74136507 \approx 0.7413$$

$$\frac{d\,out\,y_1}{d\,y_1} = \frac{d\left(\frac{1}{1+e^{-y_1}}\right)}{d\,y_1} \Rightarrow out\,y_1(1 - out\,y_1)$$

$$= 0.7513(1 - 0.7513)$$

$$= 0.1868$$

$$\frac{d\,S(x)}{dx} = S($$

$$\frac{dy_1}{dw_5} = \frac{d(out H_1 * w_5 + out H_2 * w_6 + b_2)}{dw_5} = out H_1 = 0.5932$$

$$\frac{dE_{total}}{dw_5} = \frac{dE_{total}}{dout y_1} * \frac{dout y_1}{dy_1} * \frac{dy_1}{dw_5}$$

$$= 0.0821 \cdot (\text{change in } w_5)$$

updating w_5

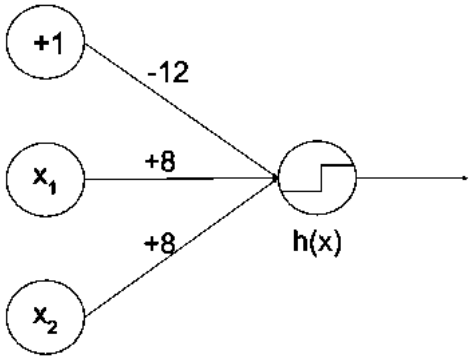
$$w_5(new) = w_5(old) - \eta * \frac{dE_{total}}{dw_5}$$

η = learning rate = 0.5

$$= 0.4 - 0.5 \times 0.0821$$

$$\boxed{= 0.3581}$$

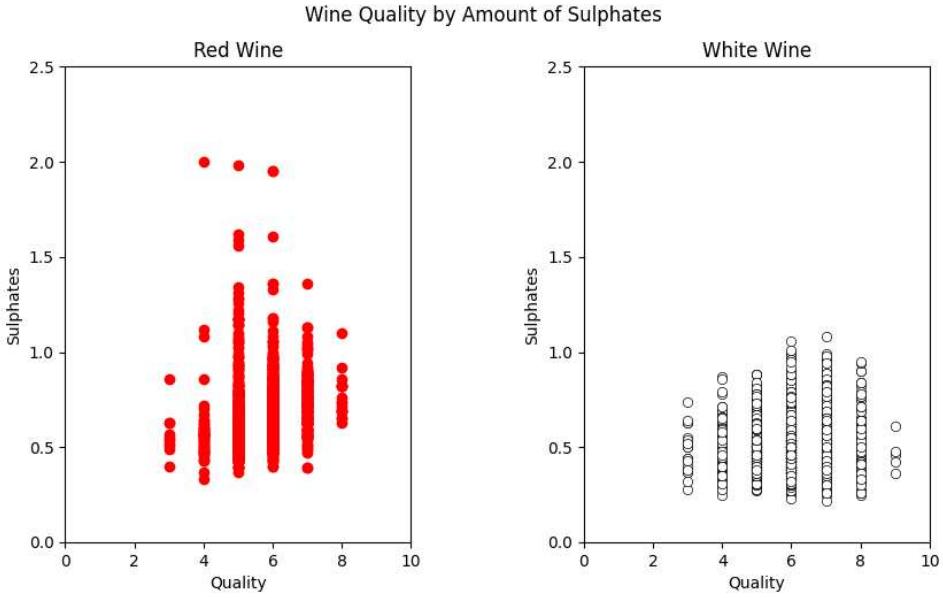
(f) The following neural networks which take two binary valued inputs x_1 , $x_2 \in \{0, 1\}$ and the activation function is the threshold function (h(x) = 1 if x > 0; 0 otherwise). Which logical functions does the network compute? (5 points)



Answer: AND logic gate (x1 = 0, x2=0 → h(x) = 0; x1 = 1, x2=0 → h(x) = 0; x1 = 1, x2=1 → h(x) = 0; x1 = 1, x2=1 → h(x) = 1)

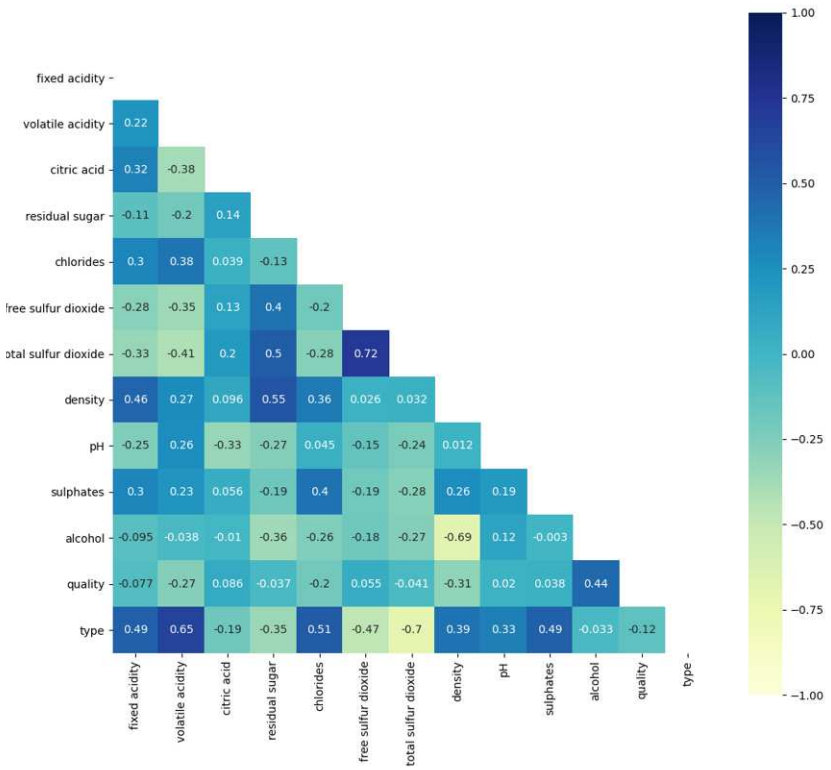
2. **Exercise 4.2: Programming: Multilayer perception classification [50 points]**
Download data files (red-wine.csv and white-wine.csv) from supplementary. Pleases look at different python packages (keras, scikit-learn, seaborn, matplotlib.pyplot, pandas and tensorflow etc.)

(a) The sulfates is one component of wine. The sulfate can cause people to have headaches. I'm wondering if this influences the quality of the wine. Please illustrate the relation or dependency between 'sulphates' and 'quality' using figure or plot. Is there any difference in "red" and "white" wine? The figure should have axis-level and legend. (7 points)



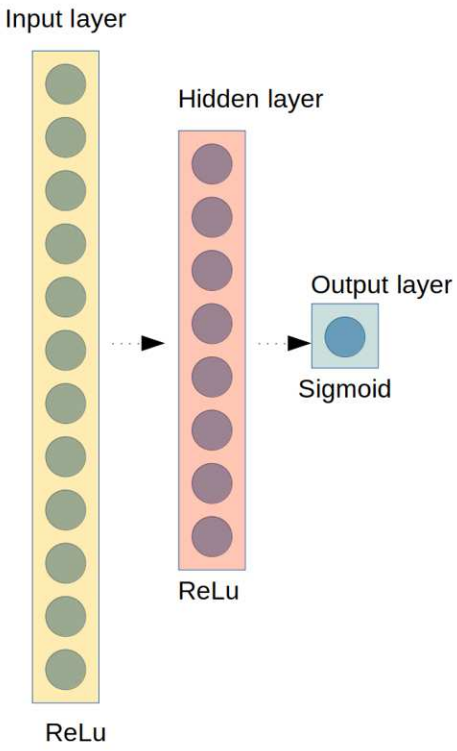
Answer: 1. High quality less sulphate. 2. white wine with a relatively low amount of sulfates that gets a score of 9. 3. the red wine seems to contain more sulfates than the white wine, which has fewer sulfates above 1 unit.

(b) Describe the correlation matrix and its important? Plot correlation matrix of variables (or features) of wine dataset. Correlation coefficient should be reflect on plot. Do you get any important information from this plot which may help you to build efficient classifier. (8 points)



Answer: 1. free sulfur dioxide and total sulfur dioxide were going to correlate, no point to use both variable in classification 2. volatile acidity and type are more closely connected, therefore "volatile acidity" is an important feature fro classification

(c) Build a neural network architecture as shown in figure below to predict the class of wine (red = '1' and white = '0'). Use 5-fold cross validation. Present accuracy from each fold and evaluate the performance of your final model with accuracy, F1-score, recall, precision. Standardized the wine data (zero mean and unit variable) before classification. Set batch size and epoches to 10 and 150, receptively. Use binary cross entropy loss as loss function and ADAM optimizer. (25 points)



Answer:

```
>>>>  Folds evaluation >>>>

Fold #1
Fold score (accuracy): 0.9738461375236511

Fold #2
Fold score (accuracy): 0.9815384745597839

Fold #3
Fold score (accuracy): 0.9915319681167603

Fold #4
Fold score (accuracy): 0.988452672958374

Fold #5
Fold score (accuracy): 0.988452672958374

#####  Evaluation of Final Model #####

Final score (accuracy): 0.988452672958374
Final score (Precision): 0.9851228978007762
Final score (Recall): 0.9524702939337085
Final score (F1_score): 0.9685214626391097
```

(d) What is your opinion on the wine dataset? Is there any way to improve the performance of your model using some data processing methods? If yes, then please provide the performance (accuracy,f1-score,recall,precision) of the improved model and explain the reason behind the improvement. (10 points)

Answer:

```
SMOTE

>>>>  Folds evaluation >>>>

Fold #1
Fold score (accuracy): 0.9811224341392517

Fold #2
Fold score (accuracy): 0.9836651086807251

Fold #3
Fold score (accuracy): 0.9831546545028687

Fold #4
Fold score (accuracy): 0.9892802238464355

Fold #5
Fold score (accuracy): 0.9872384071350098

#####  Evaluation of Final Model #####

Final score (accuracy): 0.9872384071350098
Final score (Precision): 0.9856850715746421
Final score (Recall): 0.9840751327072275
Final score (F1_score): 0.9848794442174091
```

```
# import
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import KFold
from scipy.stats import zscore
from sklearn import metrics
from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_scor
e, cohen_kappa_score
import imblearn
from imblearn.over_sampling import SMOTE

# Read in white wine data
white=pd.read_csv("white.csv",sep = ";")

# Read in red  wine data
red=pd.read_csv("red.csv",sep = ";")

# ******* 4.2 (a)

# The sulfates is one component of wine. The sulfate can cause people to have headac
hes. I'm wondering if this influences the quality of the wine. Please illustrate the
 relation or dependency between 'sulphates' and 'quality' using figure or plot. Is t
here any difference in "red" and "white" wine?

fig, ax = plt.subplots(1, 2, figsize=(8, 4))

ax[0].scatter(red['quality'], red["sulphates"], color="red")
ax[1].scatter(white['quality'], white['sulphates'], color="white", edgecolors="black
", lw=0.5)

ax[0].set_title("Red Wine")
ax[1].set_title("White Wine")
ax[0].set_xlabel("Quality")
ax[1].set_xlabel("Quality")
ax[0].set_ylabel("Sulphates")
ax[1].set_ylabel("Sulphates")
ax[0].set_xlim([0,10])
ax[1].set_xlim([0,10])
ax[0].set_ylim([0,2.5])
ax[1].set_ylim([0,2.5])
fig.subplots_adjust(wspace=0.5)
fig.suptitle("Wine Quality by Amount of Sulphates")

plt.show()
plt.close()


# ****** 4.2 (b)

# Describe the correlation matrix and its important?  Plot correlation matrix of fea
tures (or variable) of all  wine
# A correlation matrix is a table showing correlation coefficients between variables
.  It¿s a good idea to also do a quick data exploration, easy to interpret the realt
ion between different data.

# add class in red and white wine DataFrame
red['type'] = 1
white['type']=0
# Append 'white' to 'red'. ignore_index set to "True" because we don't want to keep
the index labels of white wine when we are appending then the data to red. We want a
 continuous indexing
wines = red.append(white, ignore_index=True)

# find correlation
corr=wines.corr()

# Generate a mask for the upper triangle of the corr matrix to represent symmetric m
```

```
atrix
mask = np.triu(np.ones_like(corr, dtype=np.bool))

sns.heatmap(corr, xticklabels=corr.columns.values,yticklabels=corr.columns.values,vm
in=-1, vmax=1,center=0, annot=True,square=True,cmap="YlGnBu",mask=mask)

#plt.ioff()
plt.show()
plt.close()

# ******* 4.2 (c)
# Specify the data
X = wines.values[:,0:11]
# Specify the target labels and flatten the array
y = np.ravel(wines.type)


##  4.2 (d)
# Oversampling transform the dataset
#oversample = SMOTE()
#X, y = oversample.fit_resample(X, y)

# Cross validation
# Use for KFold classification
kf = KFold(5, shuffle=True, random_state=42)

all_y = []
all_y_pred = []

print("\n>>>>  Folds evaluation >>>>\n")

fold = 0
for train, test in kf.split(X):
    fold+=1
    print(f"Fold #{fold}")

    x_train = X[train]
    y_train = y[train]
    x_test = X[test]
    y_test = y[test]

    #Intialize the constructor
    model = Sequential()
    # Add an input layer
    model.add(Dense(12, activation='relu', input_shape=(11,)))
    # Add one hidden layer
    model.add(Dense(8, activation='relu'))
    # Add an output layer
    model.add(Dense(1, activation='sigmoid'))

    # compile the keras model
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy']
)
    # fit the keras model on the dataset
    model.fit(x_train, y_train, validation_data=(x_test,y_test),epochs=150, batch_si
ze=10,verbose=0)

    #prediction probability model
    #y_pred = model.predict_proba(x_test)

    #prediction class
    y_pred = model.predict_classes(x_test)


    all_y.append(y_test)
    all_y_pred.append(y_pred)

    # measure this fold's RMSE, accuracy
    score_rmse = np.sqrt(metrics.mean_squared_error(y_pred,y_test))
    print(f"Fold score (RMSE): {score_rmse}")
    score = model.evaluate(x_test, y_test,verbose=1)
    print(f"Fold score (accuracy): {score[1]}")
```

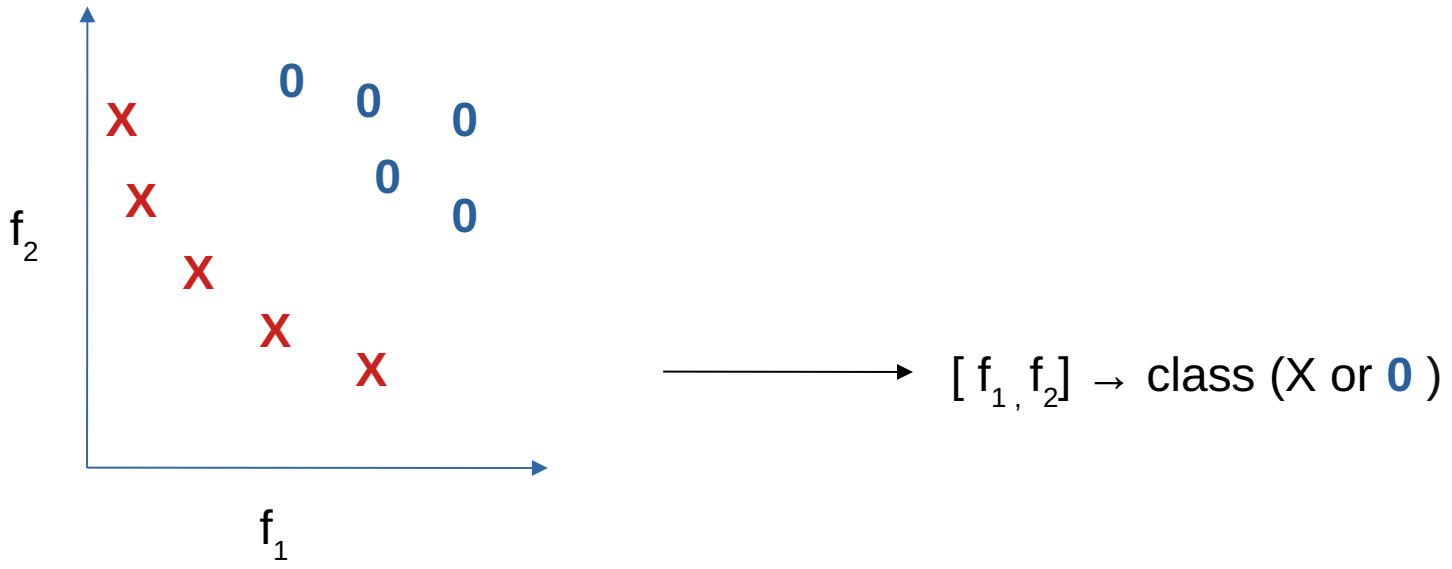
```
# 5-CV result
all_y = np.concatenate(all_y)
all_y_pred = np.concatenate(all_y_pred)
CV_score_rmse = np.sqrt(metrics.mean_squared_error(all_y_pred,all_y))
print("\n#####  Evaluation of Final Model #####\n")
print(f"Final score (RMSE): {CV_score_rmse}")
CV_score = model.evaluate(x_test, y_test,verbose=1)
print(f"Final score (accuracy): {CV_score[1]}")



# Precision
Pre_score = precision_score(all_y, all_y_pred)
print(f"Final score (Precision): {Pre_score}")
# Recall
recall = recall_score(all_y, all_y_pred)
print(f"Final score (Recall): {recall}")
#F1-score
F1 = f1_score(all_y,all_y_pred)
print(f"Final score (F1_score): {F1}")


## without CV
##Split the data up in train and test sets
#X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_st
ate=55)

## Standarization,  or mean removal and variance scaling Gaussian with zero mean and
 unit variance. The preprocessing module further provides a utility class StandardSc
aler that implements the Transformer API to compute the mean and standard deviation
on a training set so as to be able to later reapply the same transformation on the t
esting set.
## Define the scaler
#scaler = StandardScaler().fit(X_train)
## Scale the train set
#X_train = scaler.transform(X_train)
## Scale the test set
#X_test = scaler.transform(X_test)

##Intialize the constructor
#model = Sequential()
## Add an input layer
#model.add(Dense(12, activation='relu', input_shape=(11,)))
## Add one hidden layer
#model.add(Dense(8, activation='relu'))
## Add an output layer
#model.add(Dense(1, activation='sigmoid'))

## compile the keras model
#model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
## fit the keras model on the dataset
#model.fit(X_train, y_train, epochs=150, batch_size=10)
```

Assignment Sheet 4
Deep Learning
Discussion

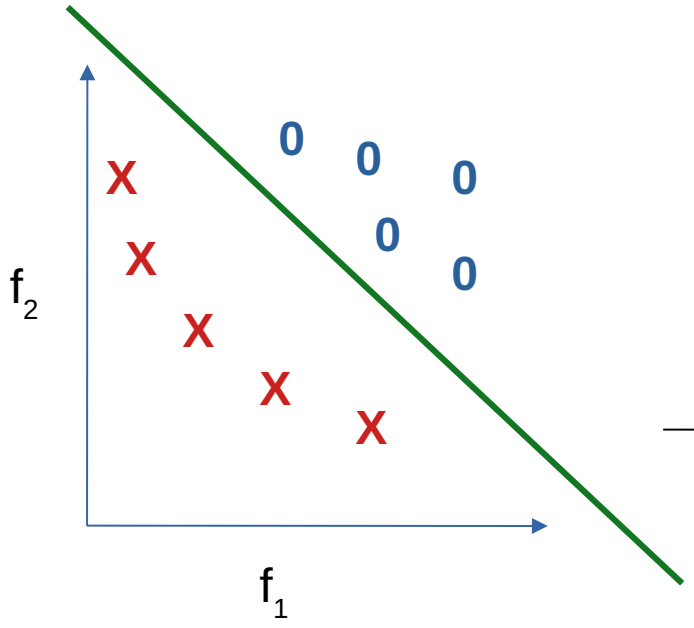
July 8th 2020

Pratiti Bhadra

(a) Briefly explain what is meant by over-fitting. How can one avoid over-fitting in deep neural networks? (5 point)



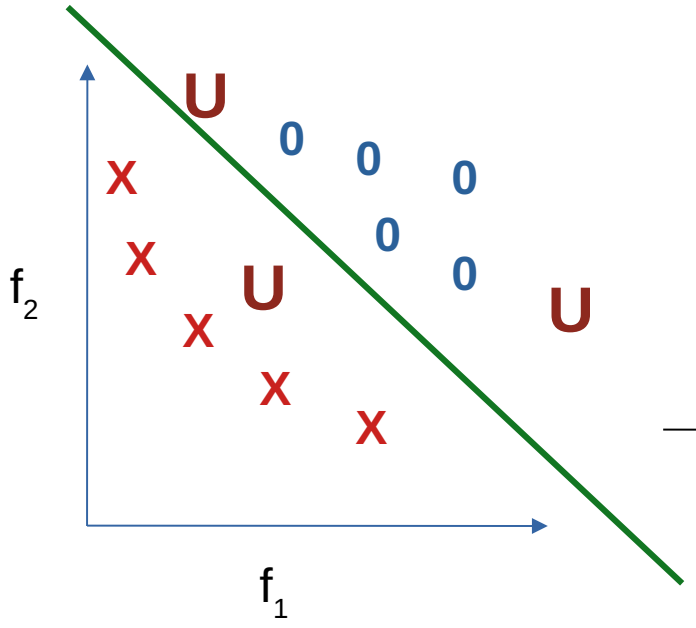
(a) Briefly explain what is meant by over-fitting. How can one avoid over-fitting in deep neural networks? (5 point)



→ [f_1 , f_2] → class (X or 0)

Decision boundary → from given data (training set)

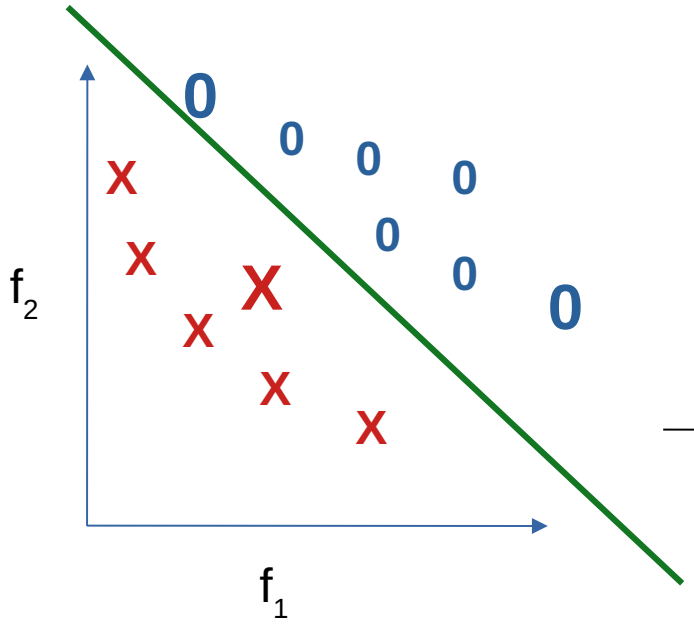
(a) Briefly explain what is meant by over-fitting. How can one avoid over-fitting in deep neural networks? (5 point)



→ [f_1 , f_2] → class (X or 0)

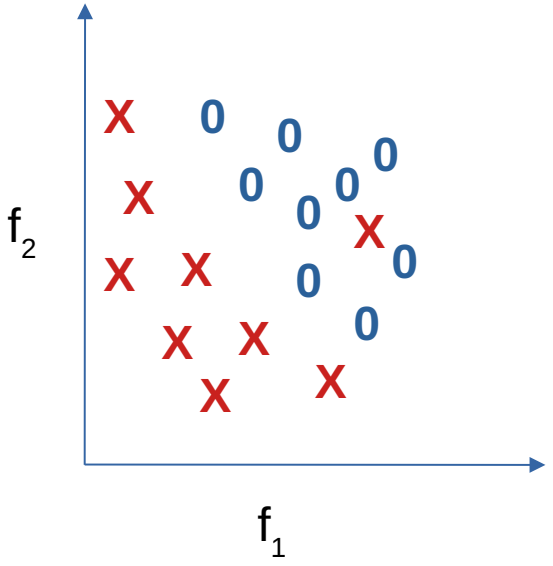
Decision boundary → from given data (training set)

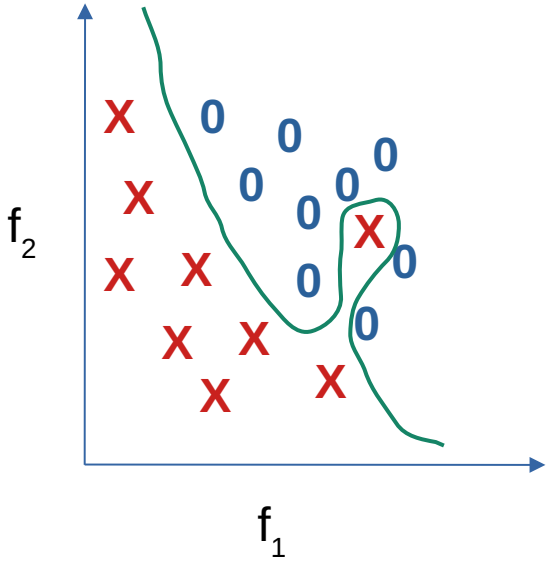
(a) Briefly explain what is meant by over-fitting. How can one avoid over-fitting in deep neural networks? (5 point)

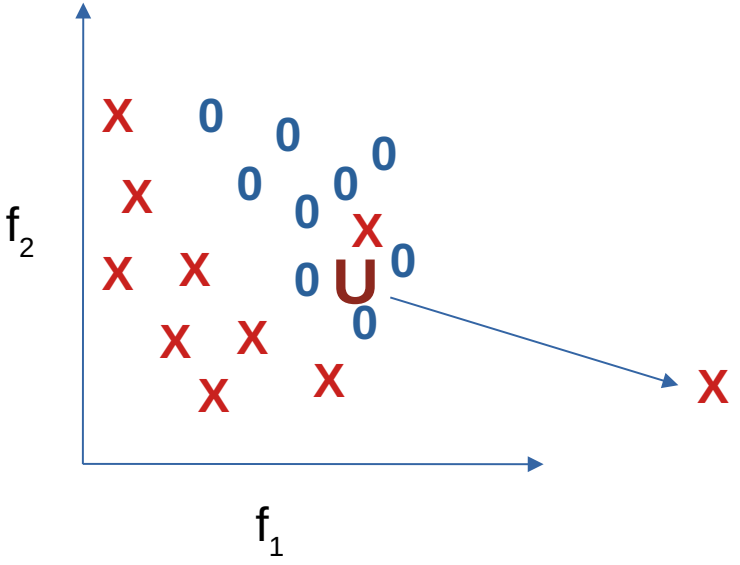


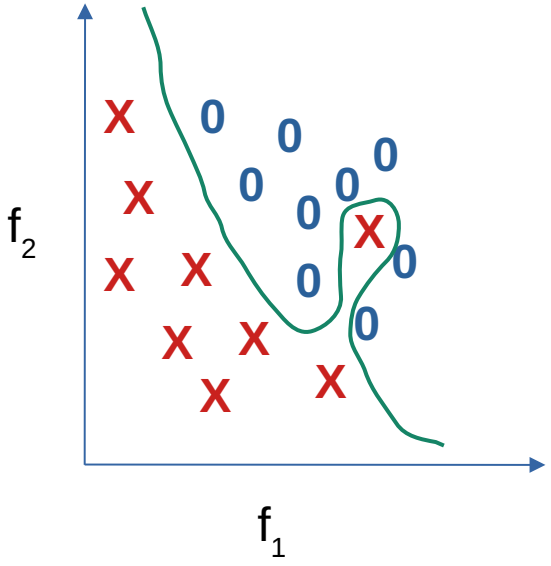
→ $[f_1, f_2]$ → class (X or 0)

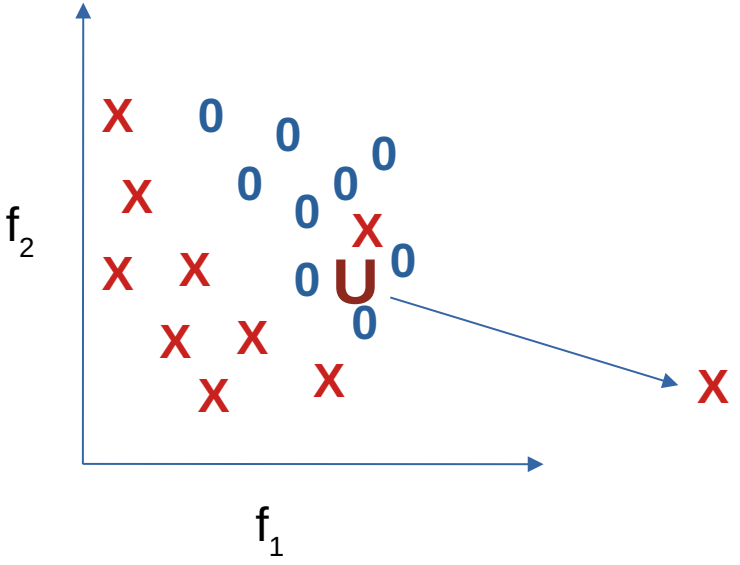
Decision boundary → from given data (training set)


f2
f1

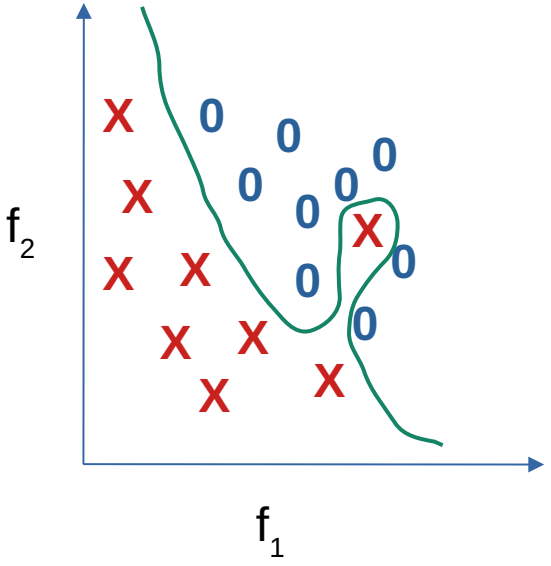

f2
f1

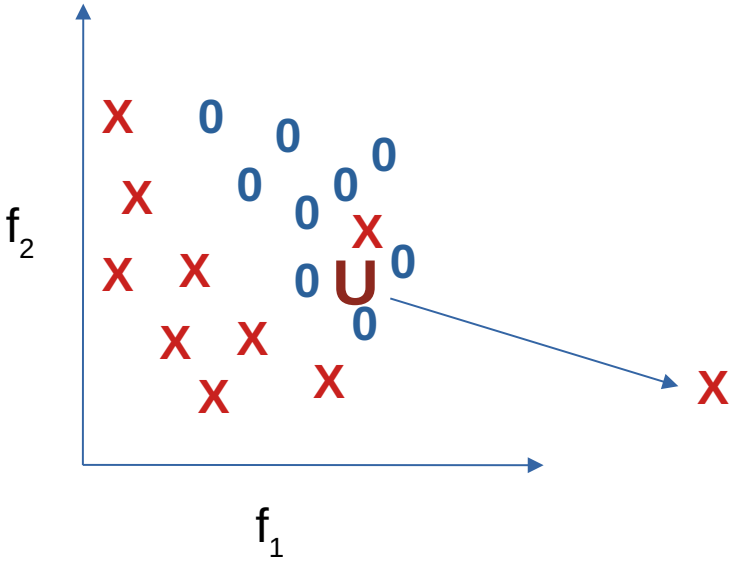

f2
U
f1





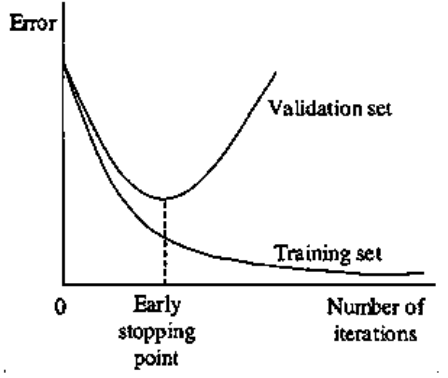
Overfitting





Overfitting

- Occurs when a model fits data too closely and therefore fails to reliably predict future observations. Error increase on test/validation data compare to training data
- In other words, overfitting occurs when a model 'mistakes' random noise for a predictable signal.
- More complex models are more prone to overfitting.



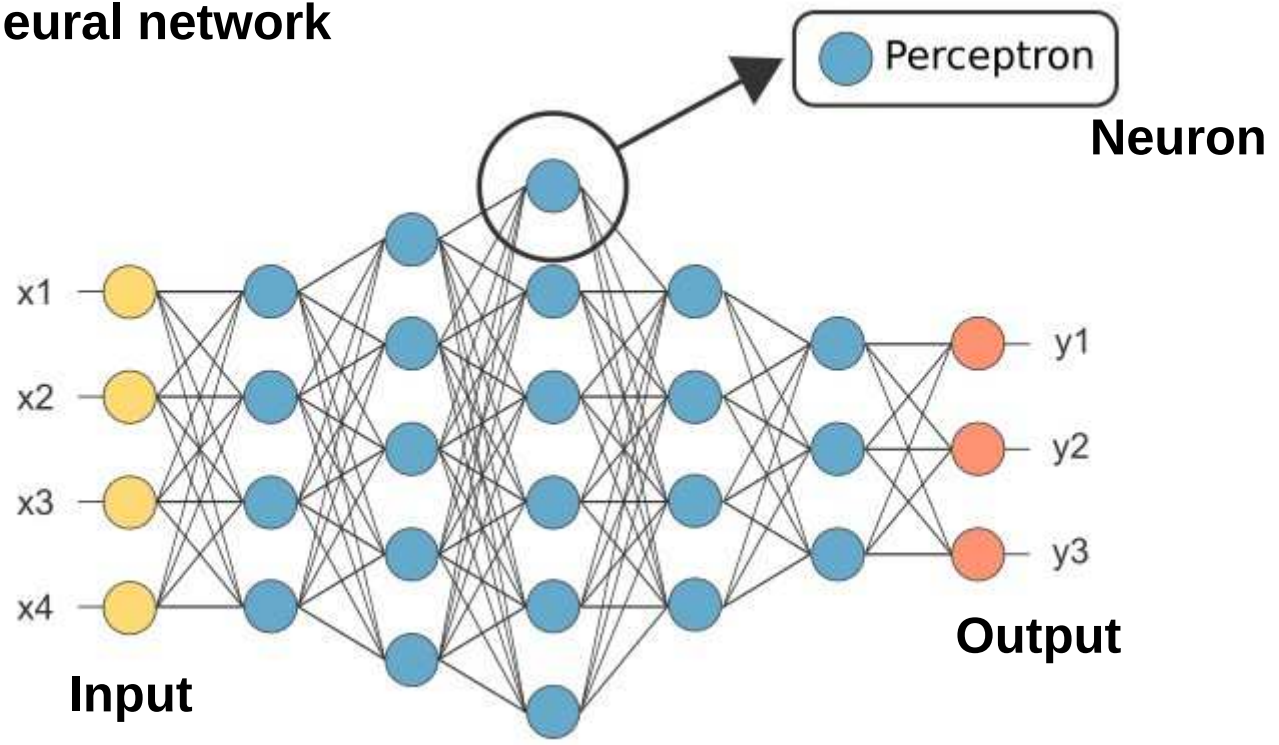
(b) What is meant by "dropout" in relation to neural networks? Which of the following statement is true for dropout? (5 points)

i. Dropout gives a way to approximate by combining many different architectures

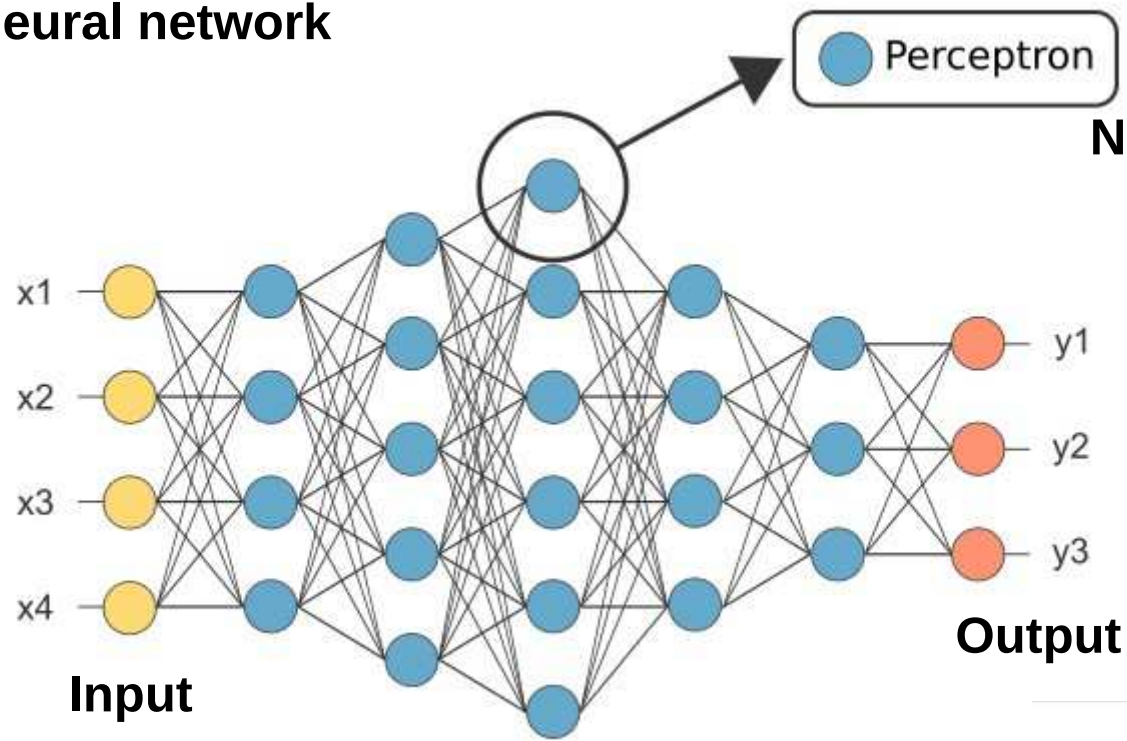
ii. Dropout can help preventing overfitting

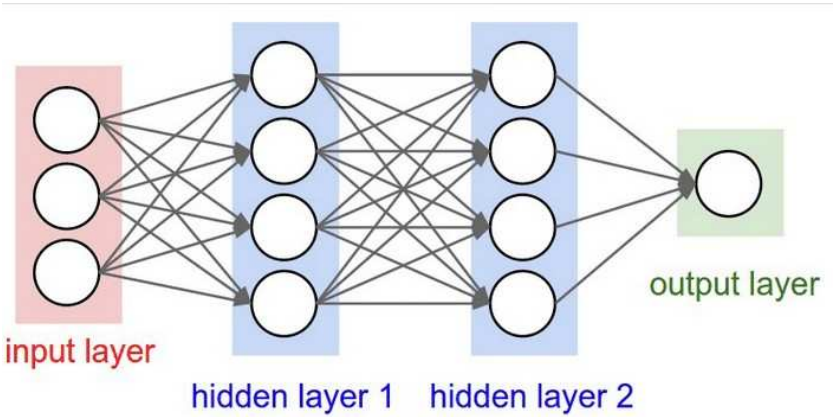
iii. Dropout prevents that the hidden layers co-adapt.

Deep neural network

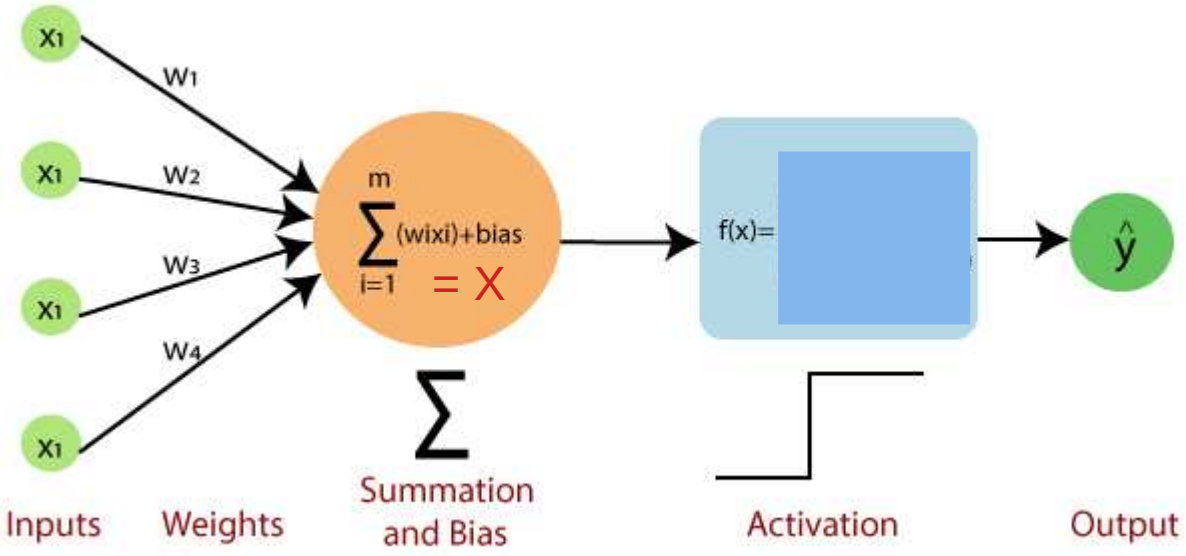


Deep neural network





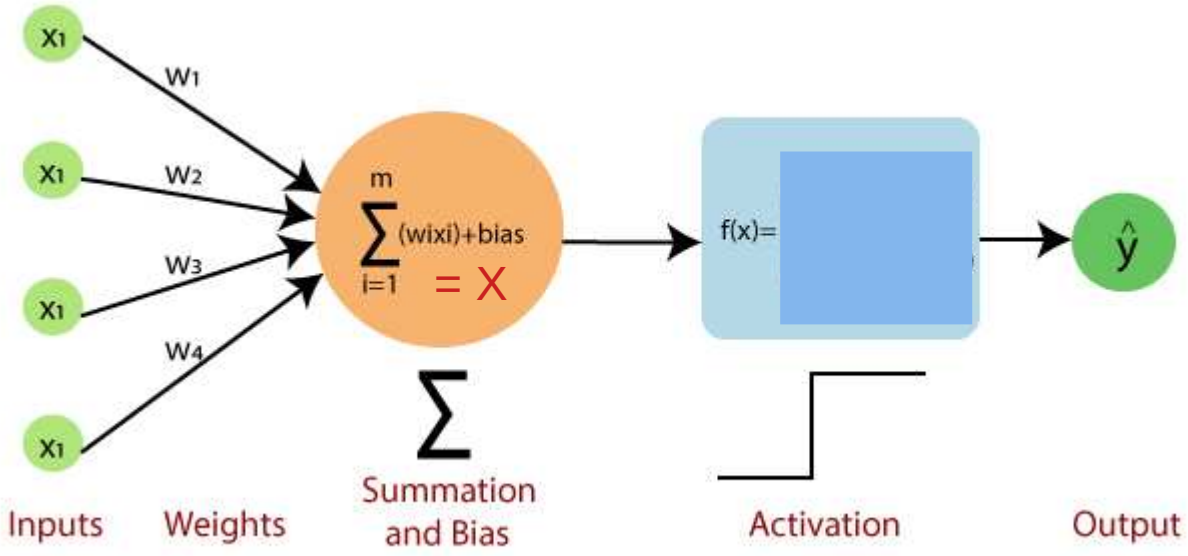
Perceptron



The perceptron consist of 4 parts

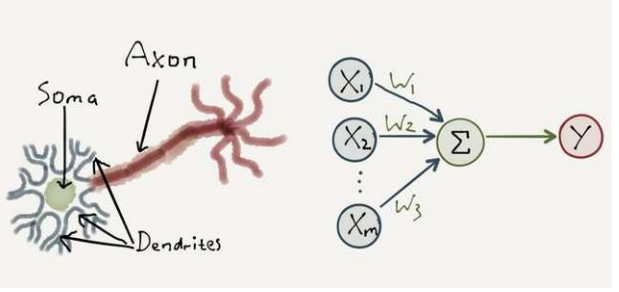
- The input value / one input layer
- Weight and Bias
- Net sub
- Activation function

Perceptron



The perceptron consist of 4 parts

- The input value / one input layer
- Weight and Bias
- Net sub
- Activation function



(c) What is an activation function of neural networks? What is the purpose of the activation function in neural networks? What are the advantages of ReLu function over Sigmoid function? (5 points)

An **activation function** determines the output behavior of each node, or neuron in an artificial neural network.

(c) What is an activation function of neural networks? What is the purpose of the activation function in neural networks? What are the advantages of ReLu function over Sigmoid function? (5 points)

An **activation function** determines the output behavior of each node, or neuron in an artificial neural network.

The purpose of the activation function is to introduce **non-linearity** into the output of a neuron

(c) What is an activation function of neural networks? What is the purpose of the activation function in neural networks? What are the advantages of ReLu function over Sigmoid function? (5 points)

An **activation function** determines the output behavior of each node, or neuron in an artificial neural network.

The purpose of the activation function is to introduce **non-linearity** into the output of a neuron

$$\sum_{i=1}^{m}(w_i x_i) + bias$$

(c) What is an activation function of neural networks? What is the purpose of the activation function in neural networks? What are the advantages of ReLu function over Sigmoid function? (5 points)

An **activation function** determines the output behavior of each node, or neuron in an artificial neural network.

The purpose of the activation function is to introduce **non-linearity** into the output of a neuron

$$\sum_{i=1}^{m}(w_i x_i) + bias$$

→ **Ax + b**

Function of a line
Y = mx + b

(c) What is an activation function of neural networks? What is the purpose of the activation function in neural networks? What are the advantages of ReLu function over Sigmoid function? (5 points)

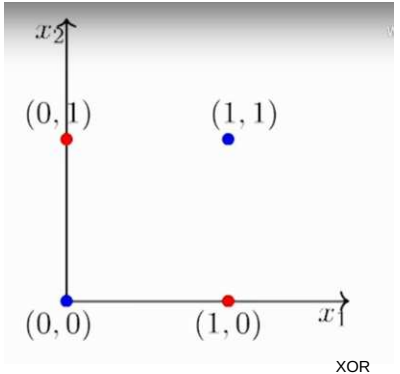
An **activation function** determines the output behavior of each node, or neuron in an artificial neural network.

The purpose of the activation function is to introduce **non-linearity** into the output of a neuron

$$\sum_{i=1}^{m}(w_i x_i) + bias$$

Ax + b

Function of a line
Y = mx + b



XOR

(c) What is an activation function of neural networks? What is the purpose of the activation function in neural networks? What are the advantages of ReLu function over Sigmoid function? (5 points)

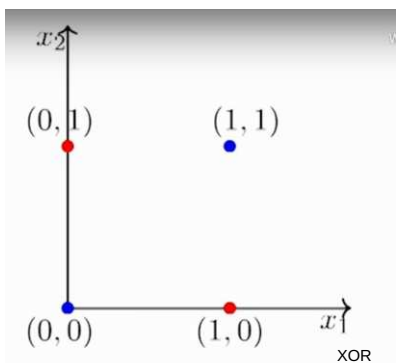
An **activation function** determines the output behavior of each node, or neuron in an artificial neural network.

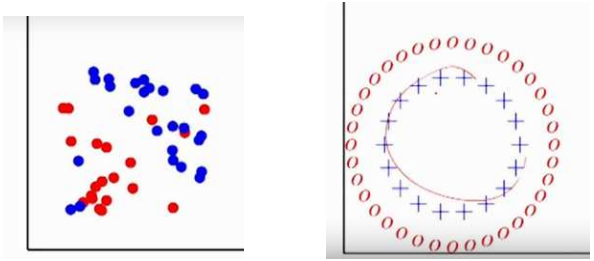
The purpose of the activation function is to introduce **non-linearity** into the output of a neuron

$$\sum_{i=1}^{m}(w_i x_i) + bias$$

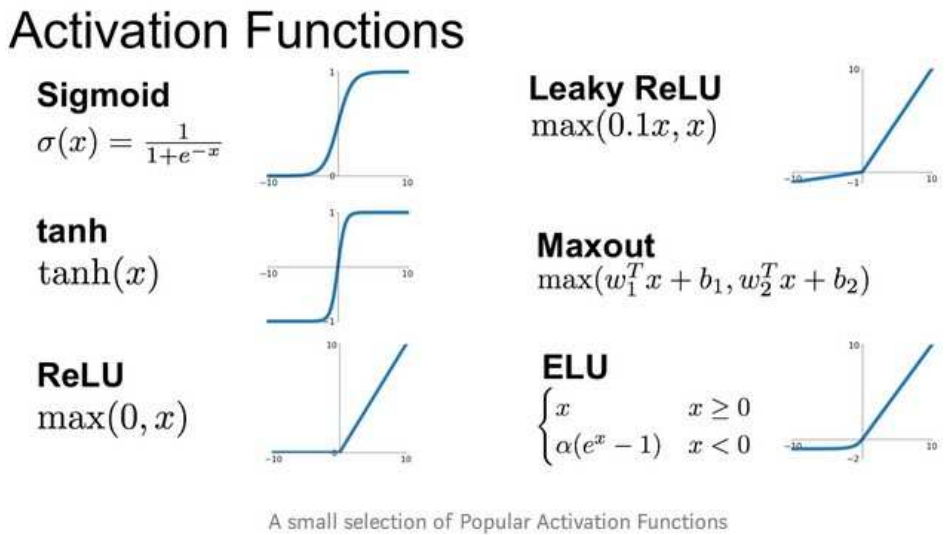
Ax + b

Function of a line
Y = mx + b





Real world data

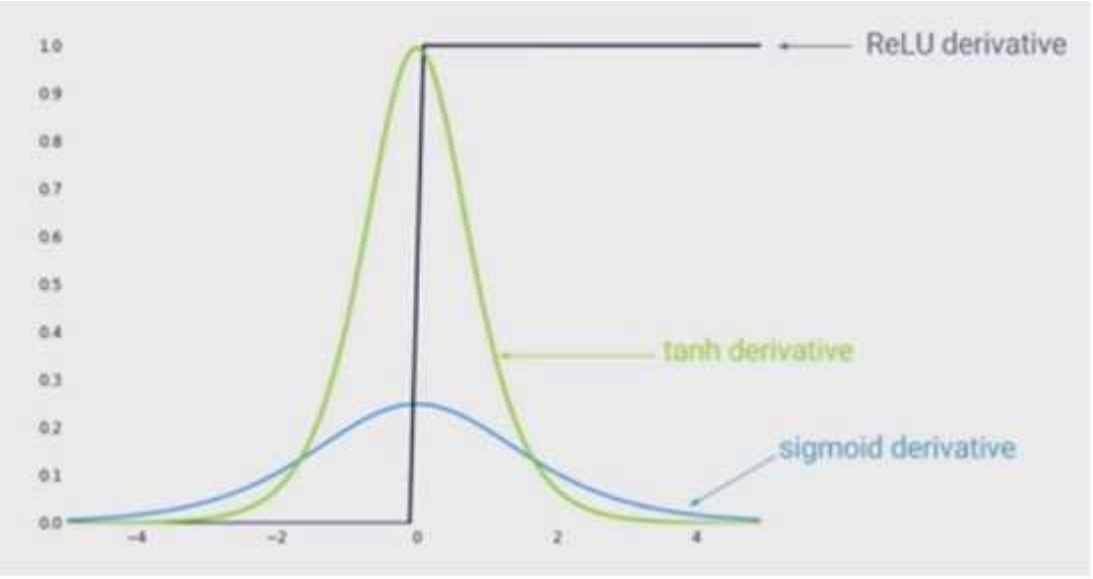


A small selection of Popular Activation Functions

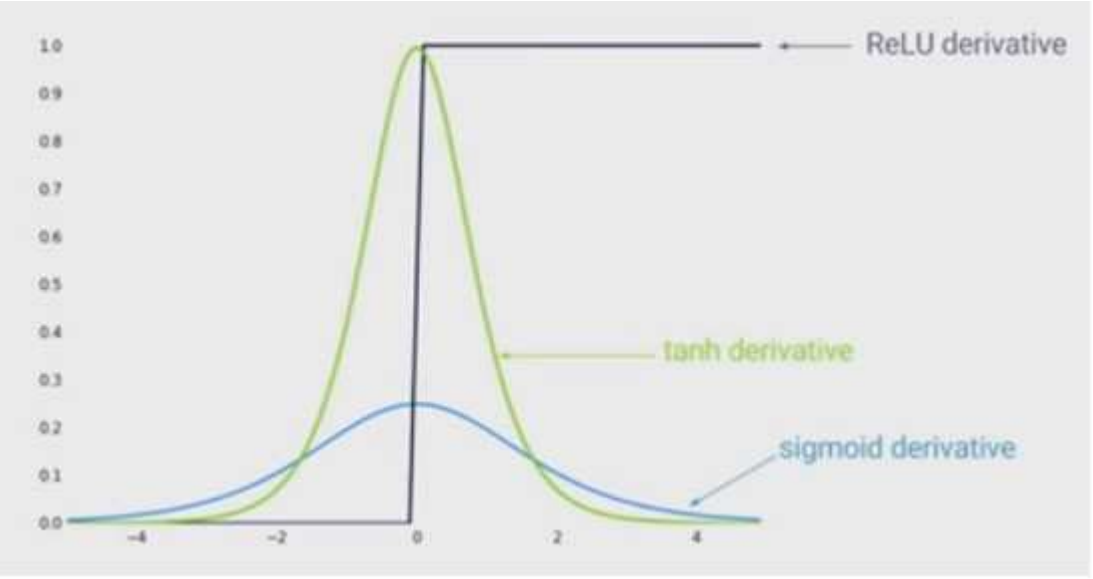
ReLU

- Computationally efficient
- Reduced likelihood of the gradient to vanish. Backpropagation technique use gradient decent to improve the performance of the neural network by updating weight.

Derivative of the function



Derivative of the function



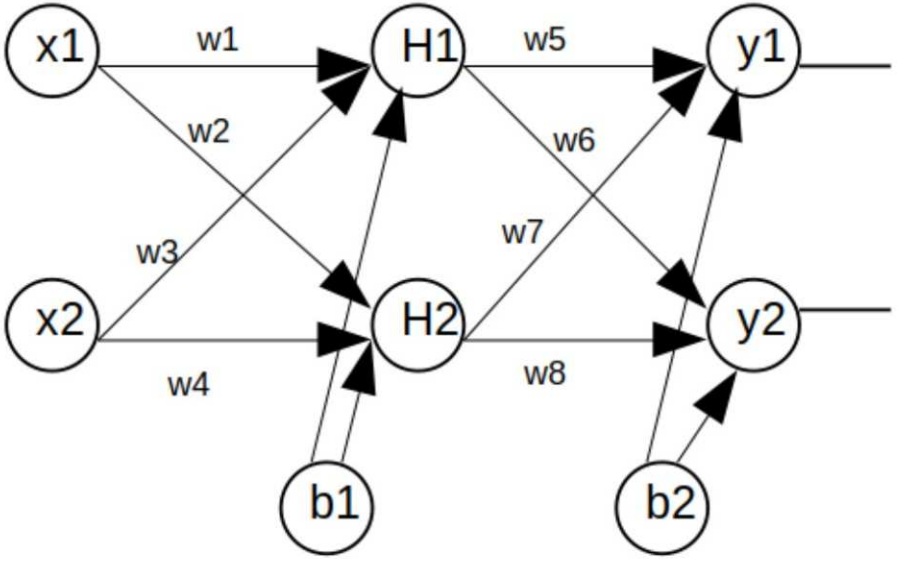
Weight update:

$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w}$$

If value of derivative is low then there will be minor change in Weight value.

It take much more time to converge in gradient desent

(e) With respect to the figure shown at the end of this problem,determine the values at hidden and output layers if x1 = 0.05, x2 = 0.10, w1 = 0.15, w2 = 0.20, w3 = 0.25, w4 =0.30, w5 = 0.40, w6 = 0.45, w7 = 0.50, w8 = 0.55, b1= 0.35 and b2=0.60. Assume that the activation function is sigmoid (logistic function $f(x) = \frac{L}{1+e^{-x}}$). Determine the total error (the mean squared error) if the target (or actual) outputs are $y1^T = 0.01$ and $y2^T = 0.99$. What is the updated weight of w5 after one iteration of the back-propagation algorithm on this example? Assume that the learning rate is 0.5. (10+5+10 = 25 points)



Forward Propagation:

$$H1 = x_1 \times w_1 + x_2 \times w_2 + b_1$$

Activation function is sigmoid = $\frac{1}{1+e^{-x}}$

output of from H1

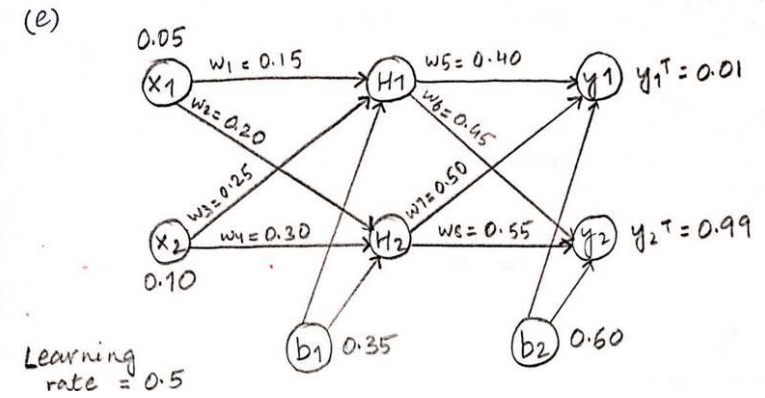
$$= \frac{1}{1+e^{-H1}}$$

(Forward Propagation)

$$H1 = 0.05 \times 0.15 + 0.10 \times 0.20 + 0.35$$

$$= 0.377$$

$$outH1 = \frac{1}{1+e^{-0.377}} = 0.593269992 \approx \boxed{0.5932}$$



Forward Propagation:

$$H1 = x_1 \times w_1 + x_2 \times w_2 + b_1$$

Activation function is sigmoid = $\frac{1}{1+e^{-x}}$

output of from H1

$$= \frac{1}{1+e^{-H1}}$$

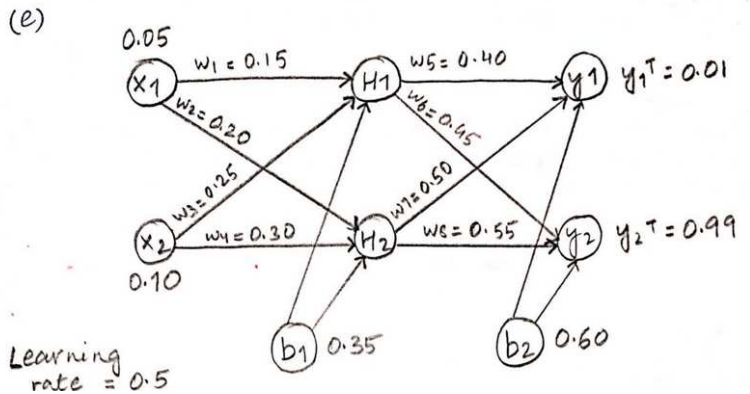
(Forward Propagation)

$$H1 = 0.05 \times 0.15 + 0.10 \times 0.20 + 0.35$$

$$= 0.377$$

$$outH1 = \frac{1}{1+e^{-0.377}} = 0.593269992 \approx \boxed{0.5932}$$

(e)



$$y1 = outH_1 * w_5 + outH_2 * w_6 + b_2 = 1.1059$$

$$outy1 = \frac{1}{1+e^{-y1}} = \frac{1}{1+e^{-1.1059}} \approx \boxed{0.7513}$$

Samway $outy2 \approx \boxed{0.7729}$

Calculate the error

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

$$= \underbrace{\frac{1}{2}(y_1^T - out_{y_1})^2}_{E_1} + \underbrace{\frac{1}{2}(y_2^T - out_{y_2})^2}_{E_2}$$

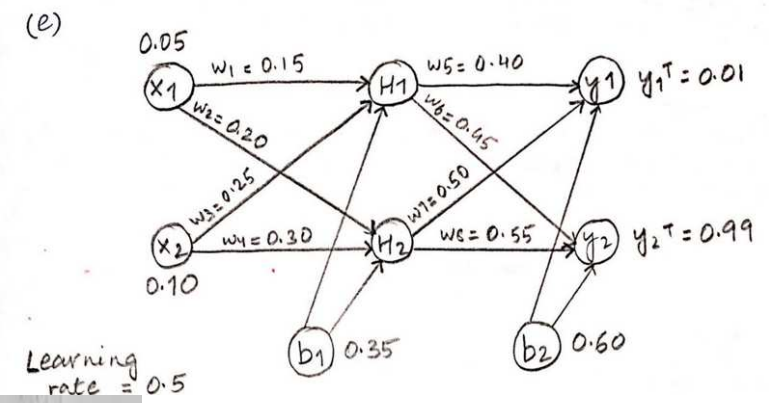
$$= 0.298371109 \approx \boxed{0.29837}$$

Backpropagation. update w_5

$$\text{Calculate error at } w_5 = \frac{\partial E_{total}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{y_1}} * \frac{\partial out_{y_1}}{\partial y_1} * \frac{\partial y_1}{\partial w_5}$$

(e)



Calculate the error

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

$$= \underbrace{\frac{1}{2}(y_1^T - out\,y_1)^2}_{E_1} + \underbrace{\frac{1}{2}(y_2^T - out\,y_2)^2}_{E_2}$$

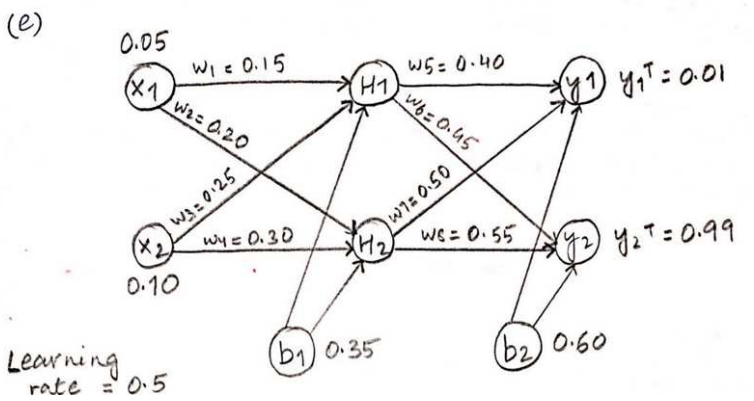
$$= 0.298371109 \approx \boxed{0.298374}$$

Backpropagation. update w_5

Calculate error at w_5 $= \frac{\partial E_{total}}{\partial w_5}$

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out\,y_1} * \frac{\partial out\,y_1}{\partial y_1} * \frac{\partial y_1}{\partial w_5}$$

(e)

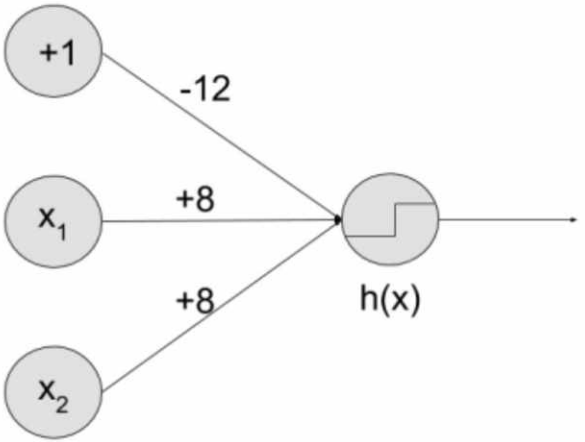


updating w_5

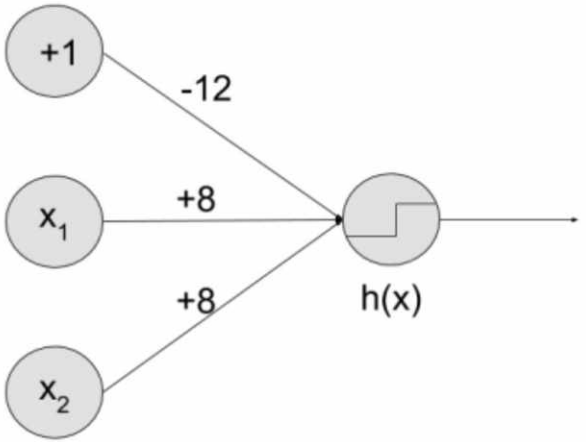
$$w_5(new) = w_5(old) - \eta * \frac{dE_{total}}{dw_5}$$

η = learning rate = 0.5

(f) Consider the following neural networks which take two binary valued inputs x_1 , $x_2 \in \{0, 1\}$ and assume that the activation function is the threshold function ($h(x) = 1$ if $x > 0$; 0 otherwise). Which logical functions does the network compute? (5 points)

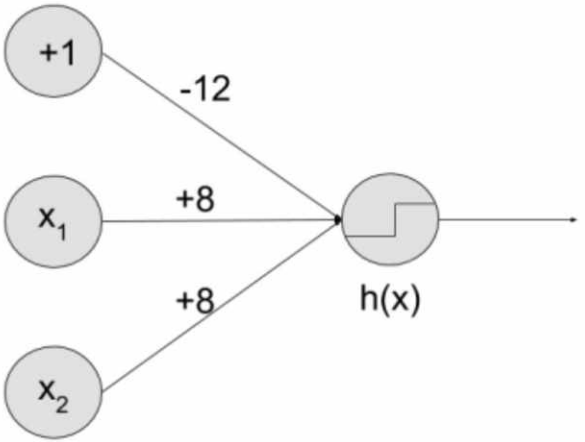


(f) Consider the following neural networks which take two binary valued inputs x_1 , $x_2 \in \{0, 1\}$ and assume that the activation function is the threshold function ($h(x) = 1$ if $x > 0$; 0 otherwise). Which logical functions does the network compute? (5 points)



Input in the output node → X1* 8 + X2 * 8 – 12 [Sum(w.x) + b]
Output of output node → if input > 0 then 1 or 0

(f) Consider the following neural networks which take two binary valued inputs x_1 , $x_2 \in \{0, 1\}$ and assume that the activation function is the threshold function (h(x) = 1 if x > 0; 0 otherwise). Which logical functions does the network compute? (5 points)



Input in the output node → X1* 8 + X2 * 8 – 12 [Sum(w.x) + b]
Output → if input > 0 then 1 or 0

Input	X1* 8 + X2 * 8 – 12	**Output**
X1 = 0 and X2 =0	0*8 + 0*8 -12 = -12	0
X1 = 0 and X2 =1	0*8 + 1*8 -12 = -4	0
X1 = 1 and X2 =0	1*8 + 0*8 -12 = -4	0
X1 = 1 and X2 =1	1*8 + 1*8 -12 = +4	1

→ **AND logic**

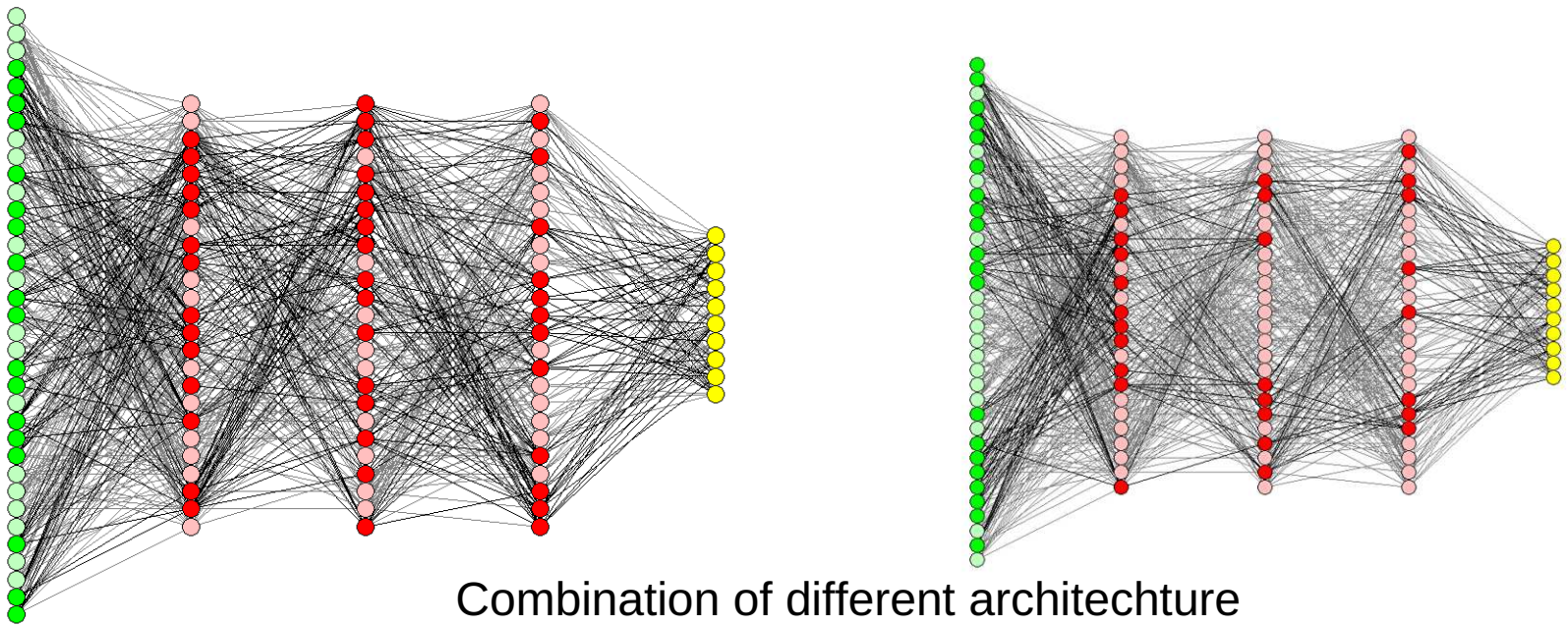
(b) What is meant by "dropout" in relation to neural networks? Which of the following statement is true for dropout? (5 points)

i. Dropout gives a way to approximate by combining many different architectures

ii. Dropout can help preventing overfitting

iii. Dropout prevents that the hidden layers co-adapt.

Dropout refer to dropping out units (neuron / perceptron) randomly



Combination of different architechture

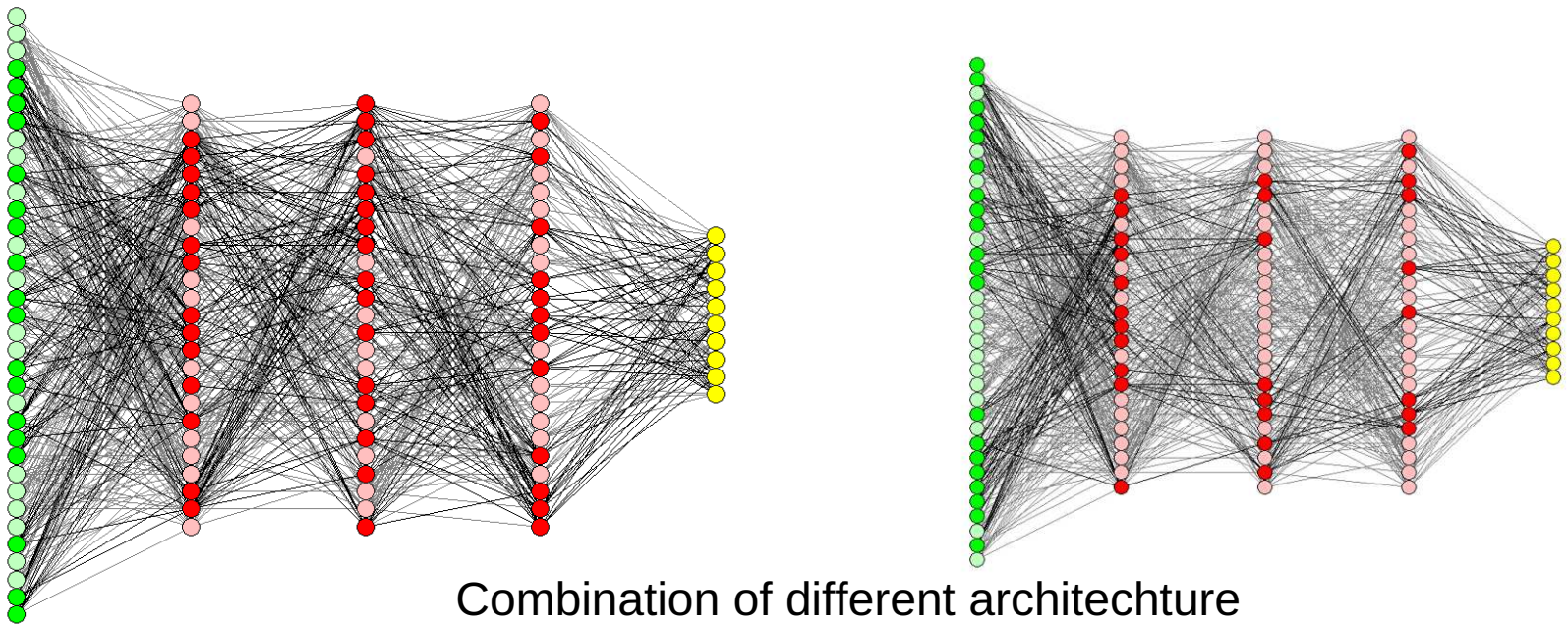
(b) What is meant by "dropout" in relation to neural networks? Which of the following statement is true for dropout? (5 points)

i. Dropout gives a way to approximate by combining many different architectures **TRUE**

ii. Dropout can help preventing overfitting **TRUE**

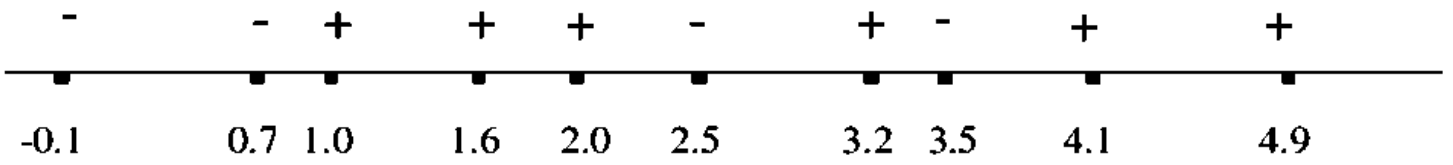
iii. Dropout prevents that the hidden layers co-adapt. **TRUE**

Dropout refer to dropping out units (neuron / perceptron) randomly



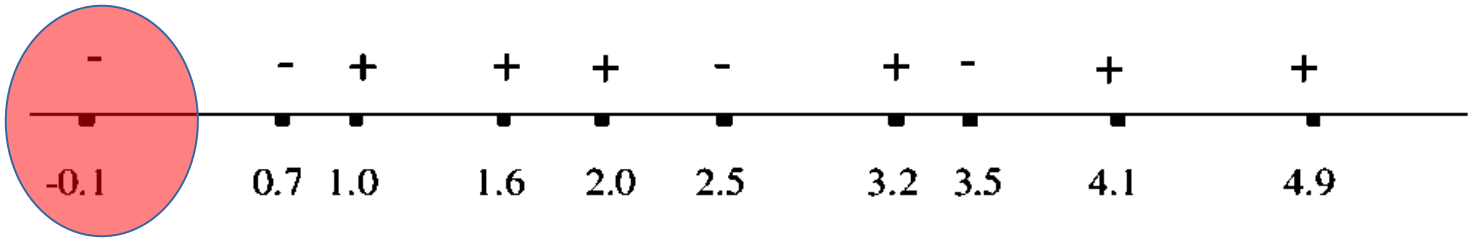
Combination of different architechture

(d) Cross validation: Carry out leave-one-out cross-validation (LOOCV) in a simple classification problem. Consider the following dataset with one real-valued input x (numbers on the line in the figure) and one binary output y (negative and positive sign). We are going to use k-NN with Euclidean distance to predict $\hat{y}$ for x.



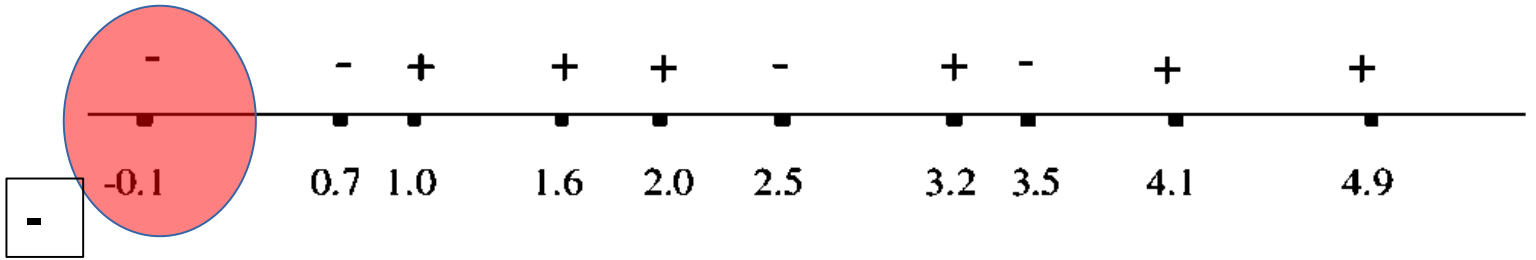
What is the LOOCV error of 1-NN on this dataset? Give your answer as the total number of misclassifications. (5 points)

(d) Cross validation: Carry out leave-one-out cross-validation (LOOCV) in a simple classification problem. Consider the following dataset with one real-valued input x (numbers on the line in the figure) and one binary output y (negative and positive sign). We are going to use k-NN with Euclidean distance to predict $\hat{y}$ for x.



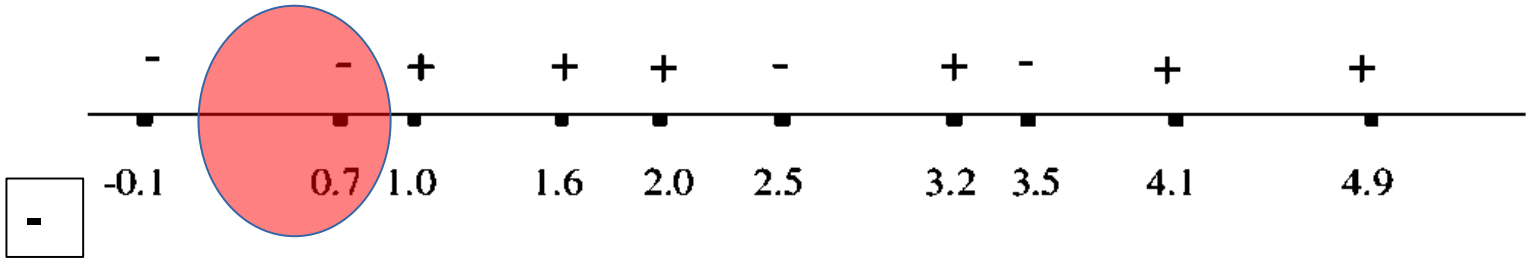
What is the LOOCV error of 1-NN on this dataset? Give your answer as the total number of misclassifications. (5 points)

(d) Cross validation: Carry out leave-one-out cross-validation (LOOCV) in a simple classification problem. Consider the following dataset with one real-valued input x (numbers on the line in the figure) and one binary output y (negative and positive sign). We are going to use k-NN with Euclidean distance to predict $\hat{y}$ for x.



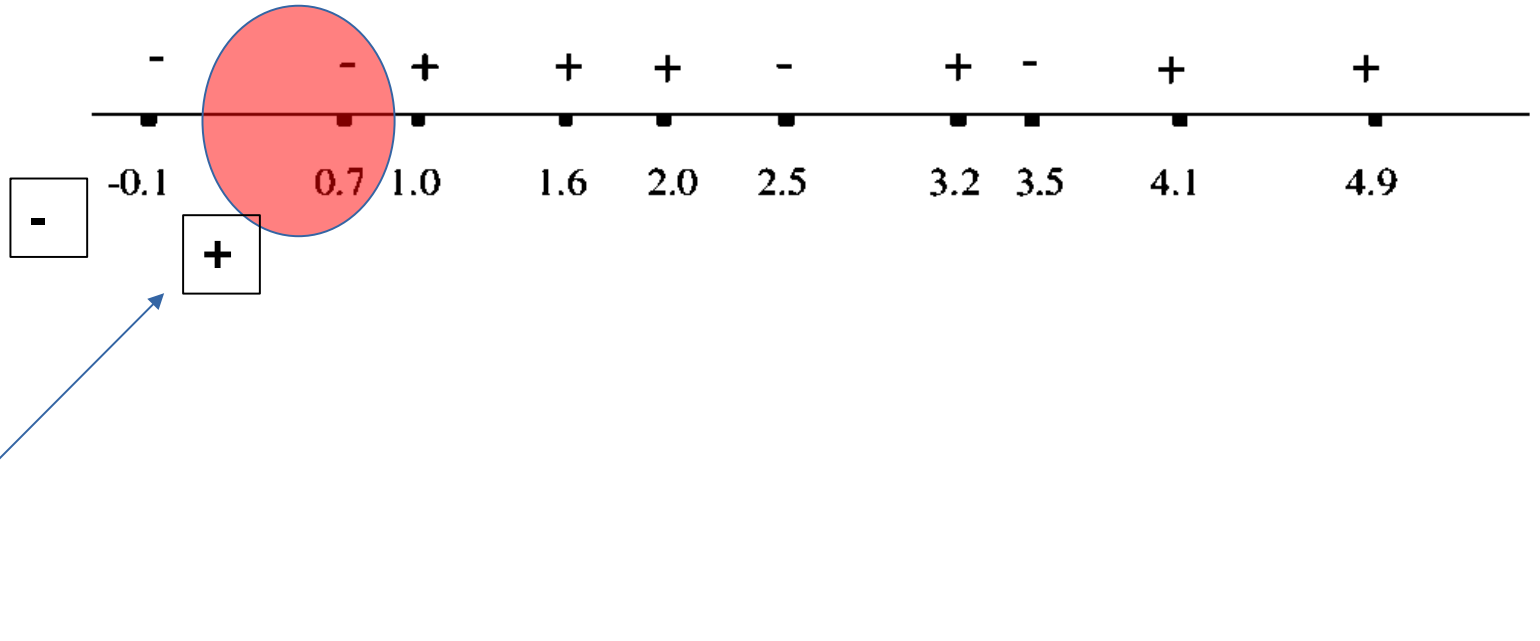
What is the LOOCV error of 1-NN on this dataset? Give your answer as the total number of misclassifications. (5 points)

(d) Cross validation: Carry out leave-one-out cross-validation (LOOCV) in a simple classification problem. Consider the following dataset with one real-valued input x (numbers on the line in the figure) and one binary output y (negative and positive sign). We are going to use k-NN with Euclidean distance to predict $\hat{y}$ for x.

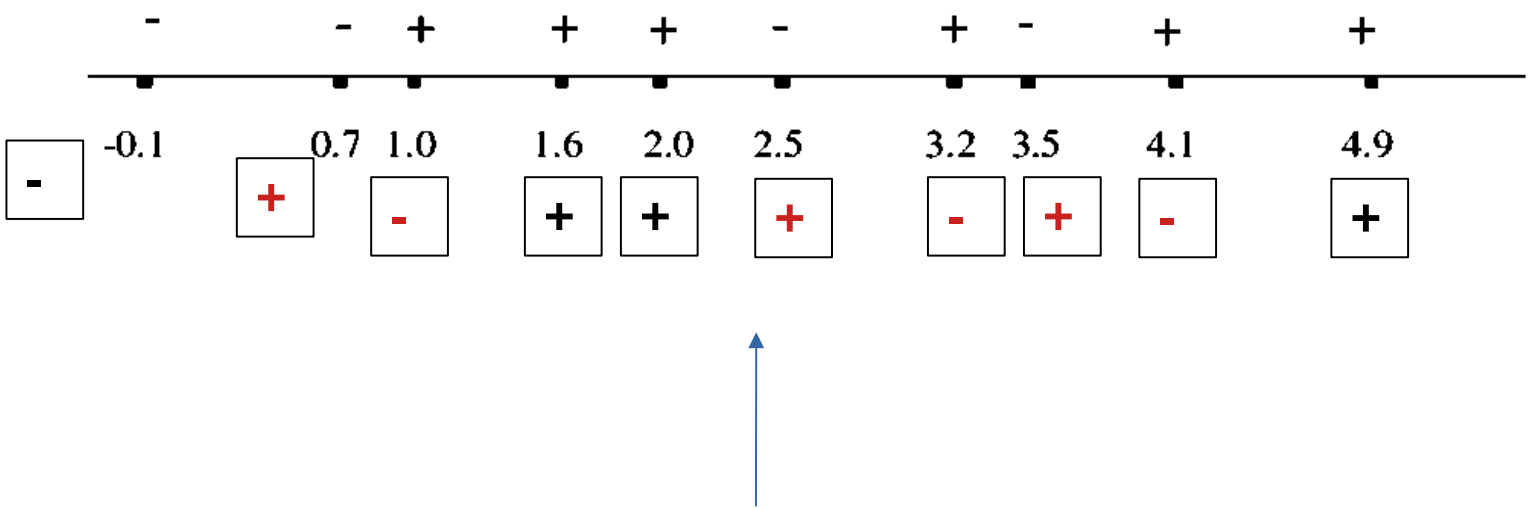


What is the LOOCV error of 1-NN on this dataset? Give your answer as the total number of misclassifications. (5 points)

(d) Cross validation: Carry out leave-one-out cross-validation (LOOCV) in a simple classification problem. Consider the following dataset with one real-valued input x (numbers on the line in the figure) and one binary output y (negative and positive sign). We are going to use k-NN with Euclidean distance to predict $\hat{y}$ for x.

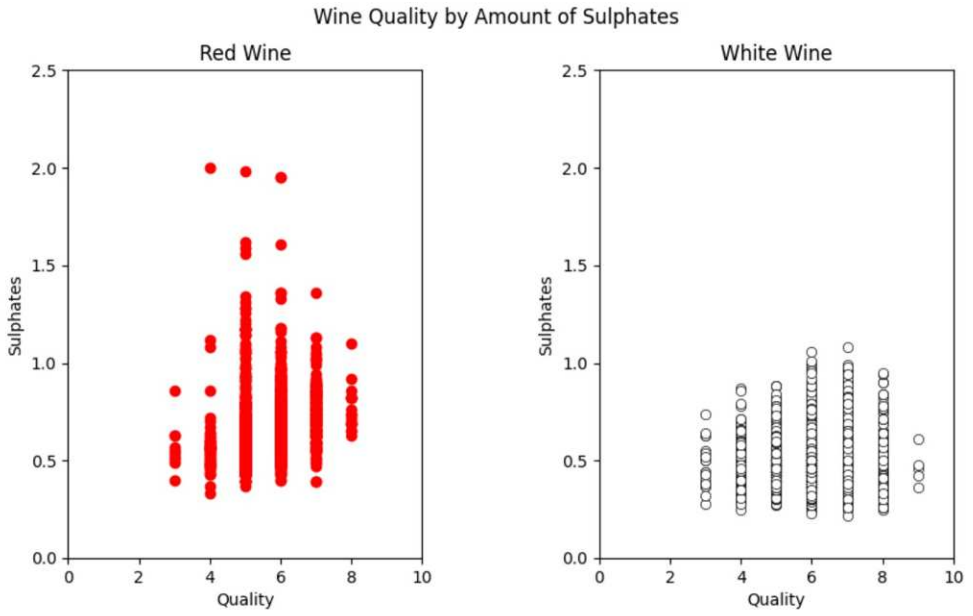


(d) Cross validation: Carry out leave-one-out cross-validation (LOOCV) in a simple classification problem. Consider the following dataset with one real-valued input x (numbers on the line in the figure) and one binary output y (negative and positive sign). We are going to use k-NN with Euclidean distance to predict $\hat{y}$ for x.



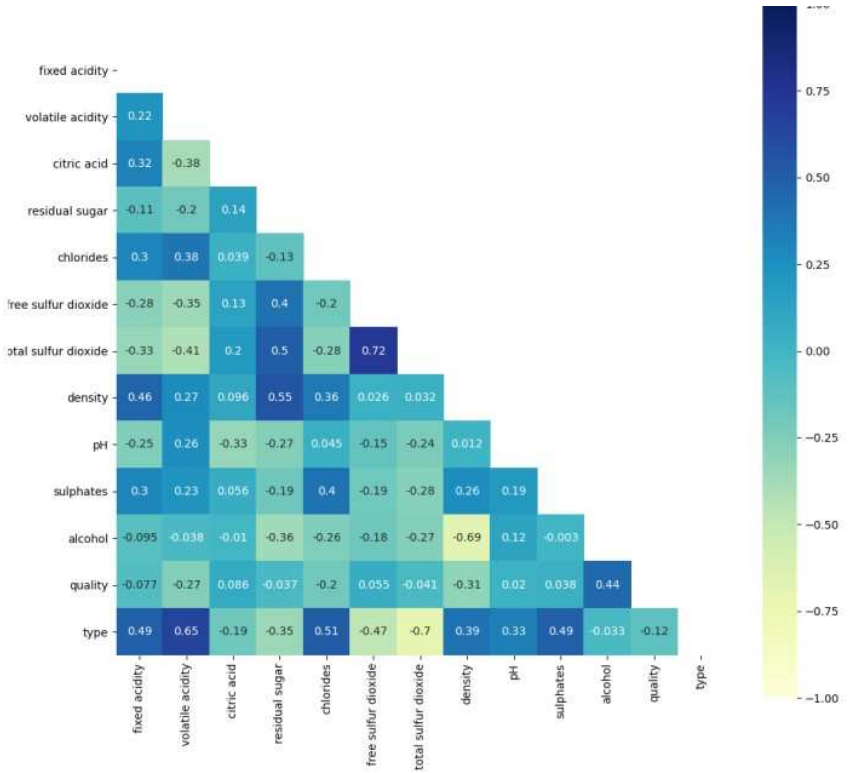
6 miss classification

(a) The sulfates are one component of wine. Sulfate ions can cause people to have headaches. I'm wondering if this influences the quality of the wine. Please illustrate the relation or dependency between 'sulphates' and 'quality' using a figure or plot. Is there any difference between "red" and "white" wine? The figure should have axis-level and legend. (7 points)



Answer: 1. High quality less sulphate. 2. white wine with a relatively low amount of sulfates that gets a score of 9. 3. the red wine seems to contain more sulfates than the white wine, which has fewer sulfates above 1 unit.

(b) Describe the correlation matrix and its important? Plot correlation matrix of variables (or features) of wine dataset. Correlation coefficient should be reflect on plot. Do you get any important information from this plot which may help you to build efficient classifier. (8 points)



Answer: 1. free sulfur dioxide and total sulfur dioxide were going to correlate, no point to use both variable in classification 2. volatile acidity and type are more closely connected, therefore "volatile acidity" is an important feature fro classification

(d) What is your opinion on the wine dataset? Is there any way to improve the performance of your model using some data processing methods? If yes, then please provide the performance (accuracy,f1-score,recall,precision) of the improved model and explain the reason behind the improvement. (10 points)

Imbalanced dataset

Imbalanced dataset can be handled by

- Oversampling
- Bagging
- ....

Oversampling

SMOTE (Synthetic Minority Oversampling TEchnique)