

Application of Neural Networks for the prediction of cancerous SNPs

**But first an example of data from
the DataBase**

RS_ID	chr	pos	ref	alt	scoreA	scoreB	Functional element	Number experiment	file_type	cell_linec	Cancer type	Cell Line cancer
rs10201930	2	189874460	G	A	0.0719	0.0659404	H3K36me3-human	1	narrow	prostate		normal
rs10178969	2	190315723	A	C	0.0657	0.0683136	EZH2-human	1	narrow	PC-3	prostate	cancer
rs10178969	2	190315723	A	C	0.0657	0.0683136	H3K36me3-human	1	narrow	PC-3	prostate	cancer
rs1037532	18	50038602	T	C	0.028	0.0344594	H3K27me3-human	1	narrow	PC-3	prostate	cancer
rs10039204	5	101180077	T	C			EZH2-human	1	narrow	PC-3	prostate	cancer
rs1002223	7	21172073	A	C	0.3313	0.272092	EZH2-human	1	narrow	PC-3	prostate	cancer
rs1002223	7	21172073	A	C	0.3313	0.272092	POLR2A-human	1	narrow	prostate gland		normal

Data - Type

RS_ID	character
Chr	class
Pos	number integer
Ref	character
Alt	character
ScoreA	number float
ScoreB	number float
functional_element	character
n_experiment	number integer
file_type	character
cell_line	character
cancer_type	character
cell_line_cancer	character

First Mission

Convert each features in integer number for the nominal data

**I need to convert
all of it in
numbers!**



Data Cleaning & Feature engineering

- **Build ScoreC = (scoreA + scoreB)/2**
 - Or use just one of the two if both are not there
 - *Example.* If not scoreB: ScoreC = scoreA
 - Drop score A and score B

Data Cleaning & Feature engineering

- **Set up data structures as dictionaries for the store of association**
 - *Theoretical example.*
 - Label_1 = Number
 - *Practical example.*
 - Functional Element | H3K36me3-human = 5
 - Cell Lines | prostate gland = 3

Data Cleaning & Feature engineering

- **Select the features for the prediction (X)**
- **Select the feature to be predicted (Y)**
- **Drop all useless feature for the prediction**

Data Cleaning & Feature engineering

- **Select the features for the prediction (X)**
- **Select the feature to be predicted (Y)**
- **Drop all useless feature for the prediction**

Features for prediction

- **Y** Target to predict = cancer_type
- **X** Using:
 - Ref , alt
 - functional_element
 - cell_line_cancer
 - scoreC

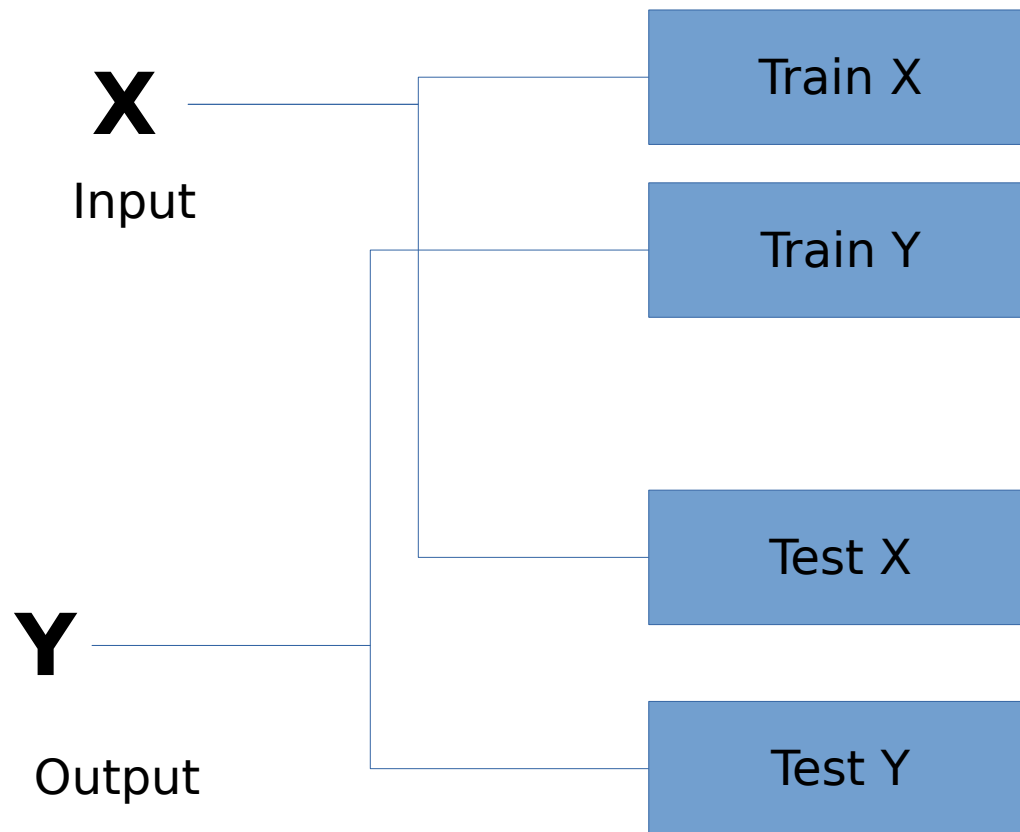
Building the model

- **Build $F(x)$ using X and Y with Neural Network Algorithms including Gradient Discendent**

$$Y = F(X)$$

- I have X and Y
- I need $F()$

SPLIT X & Y in Train & Test



Build the Neural Network model

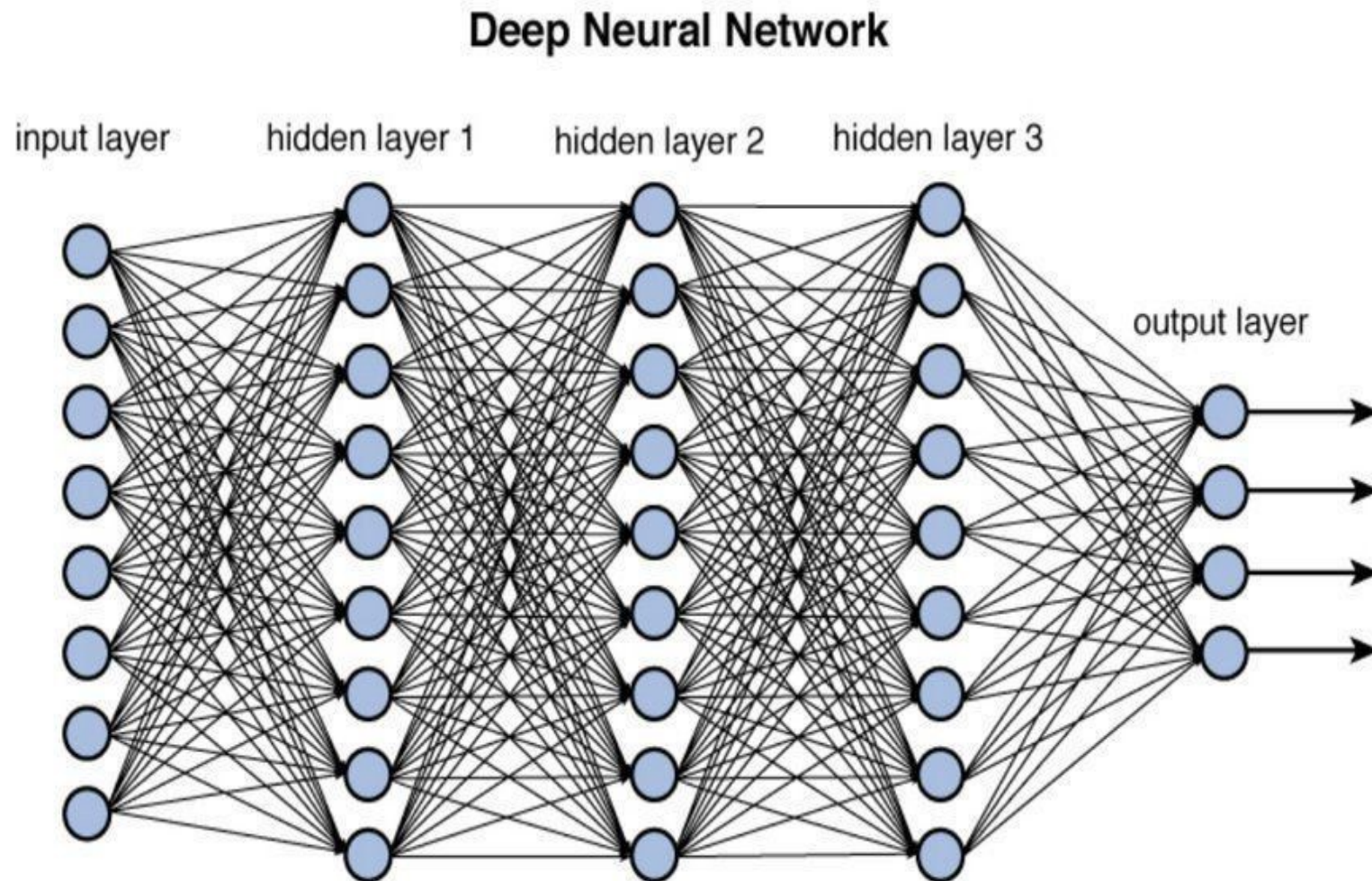
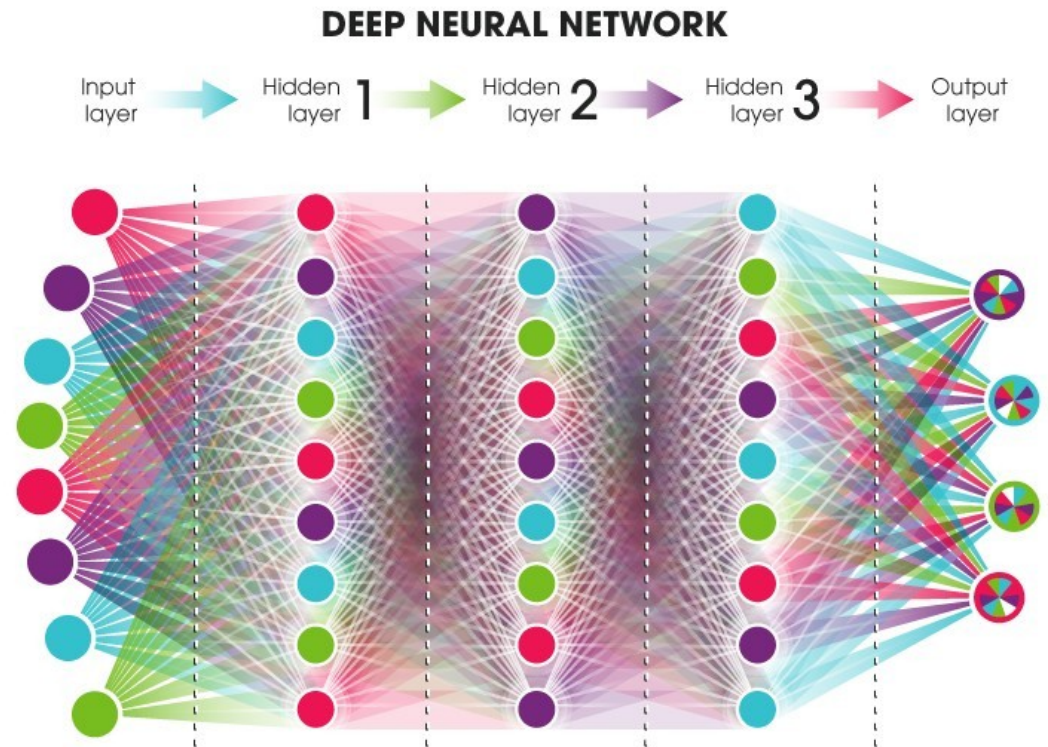


Figure 12.2 Deep network architecture with multiple layers.

Build the Neural Network model

Neural Structure

- **8 features for input**
- **16 hidden neurons**
- **32 hidden neurons**
- **16 hidden neurons**
- **1 output label**



Train & Test Data

- **Use of Train data for build the model**
- **Use of Test data for extract accuracy and predictions from the model**

KERAS for simple NN



- **Using Keras built in models**
- **`model = Sequential()`**
- **`model.compile(loss='binary_crossentropy',
optimizer='adam', metrics=['accuracy'])`**

Conclusion

- **Prediction of cancer_type label**
- **Test loss: 0.17**
- **Test accuracy: 0.87**
- **Huge changes of these values for each running time**

Conclusion

- **The script was build for serve different SNPs datasets.**
- **The script is very flexible for the same dataset. For change the entire model just select different label_target and different features used for the prediction.**