Senior Project I
XiaoEx - The Exchange Expert

By
Kasperi Reinikainen 5818014
Hein Htet Naing 5818035
Asnai Narang 5815228

Content

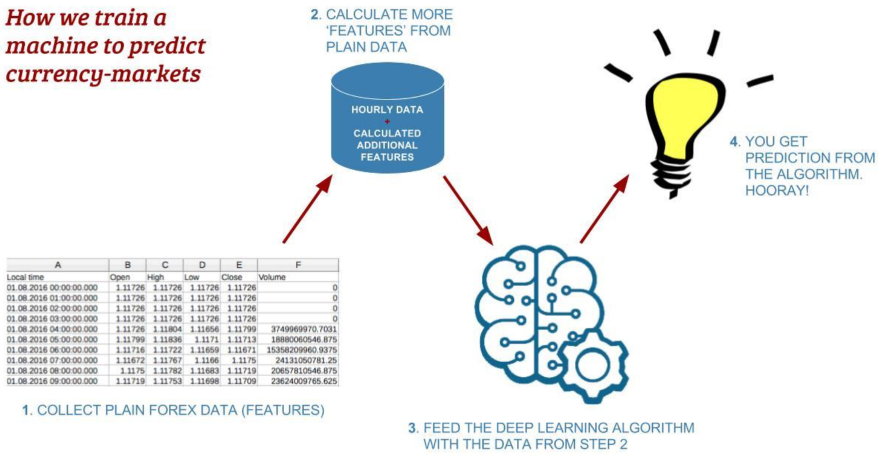
Introduction

- Given problem:
  Predict Forex market movements using DNN's

- Case-study to follow in our Initial approach

- Improve results from case-study or draw relevant conclusions of it

- Apply findings in tangible use-case



Introduction: goals

CHECKLIST:

- [ ] Comprehend and apply the Case-study’s approach in the Initial design
- [ ] Branch-off and find better models using our own techniques
- [ ] Apply the model in a tangible application use-case

Motivation and background

Main motivation
⇒ Learn and apply modern ML-techniques in a challenging use-case
⇒ Find applicability for the results

Members:

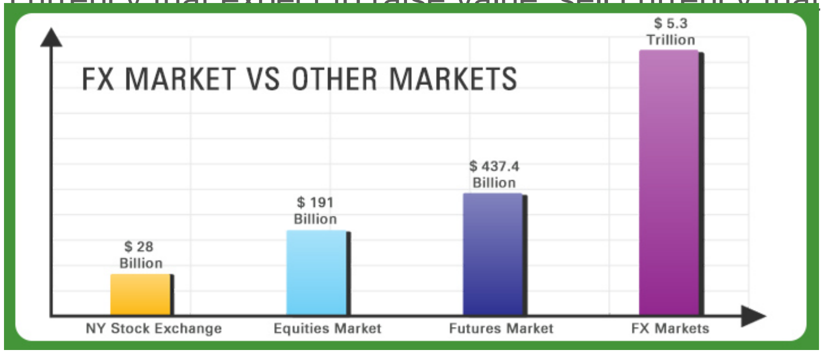

Asnai Narang, 3rd year CS major


Hein Htet Naing (Hector), 3rd year IT major


Kasperi Reinikainen, 3rd year CS major

In brief: Forex markets

- Foreign Exchange: Currency markets for trading foreign currencies in pairs
- Target users: Commercial and central banks, Investment and other large companies, Governments
- Forex trading: buy currency that expect to raise value, sell currency that is expected to lose value



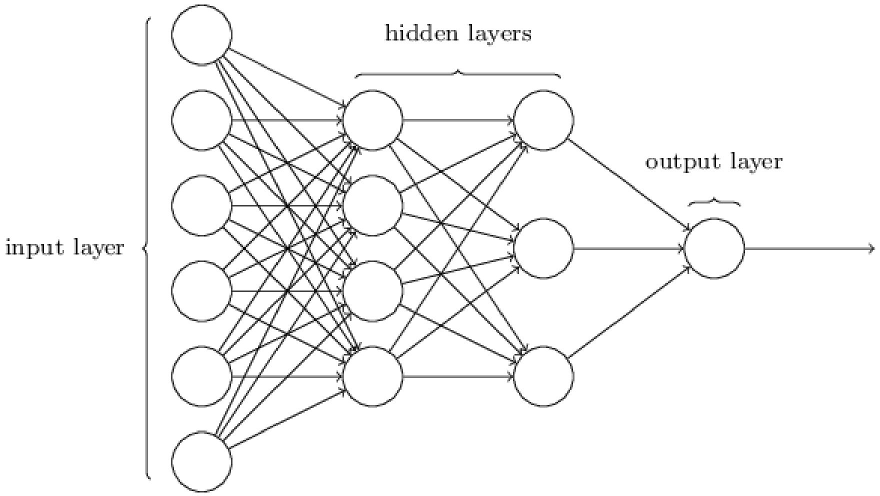
In brief: Artificial Neural Networks

- Original development inspired by Brain

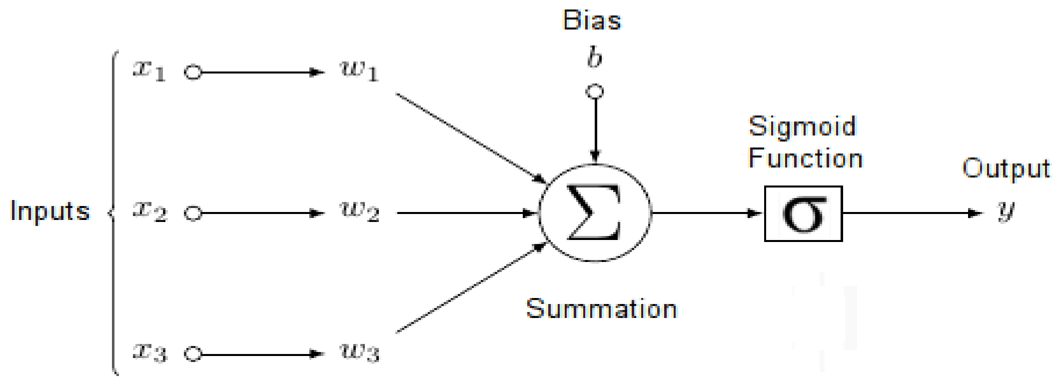
- Can potentially approximate problems with any level of complexity

- ‘Learns’ by adjusting weights between different layers of neurons

- 3 main components (not incl. loss-func.):
    1. Weight calc. (integration function)
    2. Activation function (scales the output)
    3. Optimization function (param. update)



Single neuron computational graph



Content

Reference studies

Studies regarding Forex Prediction using ML - techniques are not hard to find.

Similarities between all studies (incl. Case study):

- They all (except one using SVM) use some form of Artificial Neural Networks
- Features are pre-defined and selected mostly intuitively based on various statistical formulations of ‘raw’ OHLC - currency data
- Prediction accuracy is relatively low (ranging mostly between 40-60 % for classification problems)

Case study

Prediction of Exchange Rate Using Deep Neural Networks, presentation by University of Nagoya

Training conditions for case study:

- **Assumptions**:
  1. Future trend consists of past information.

- **Prediction types**:
  1. Classification: { Up, Down }

- **ANN Type**:
  1. Deep neural network

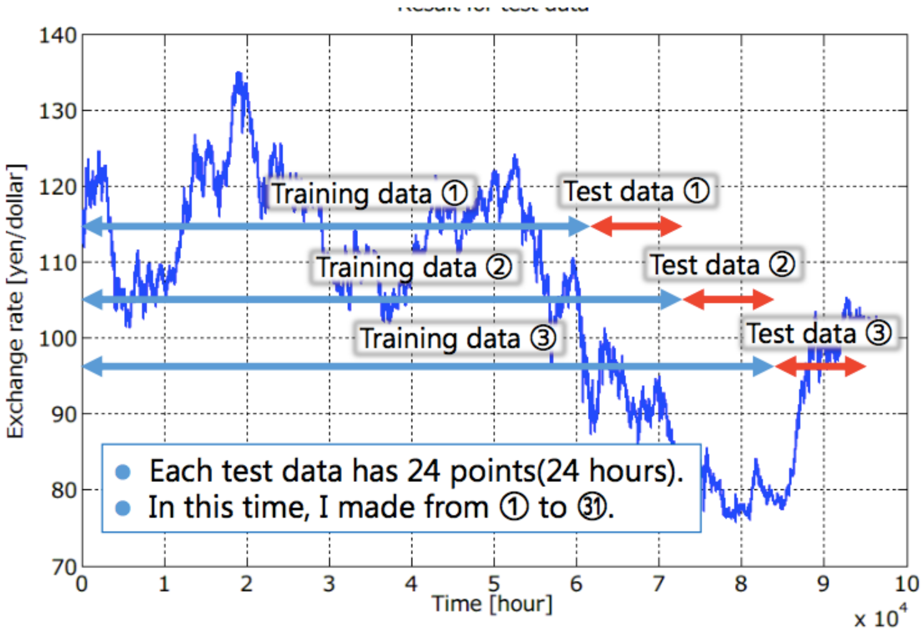
- **Features**:
  1. 10-features:
     { open, close, high, low, datetime, volume,
     RSI, stochastic RSI, Moving avg, %R }
  2. Concatenated (method unknown) to become 100 features

- **Dataset**:
  → USD/JPY 01/01/1991 - 31/12/2014
  Total of 97,362 instances

Case study training settings

T1	Instances in dataset	Train / % train	Total Features	Layers	Neurons (total)	Activation	Optimization	Learning-rate	Batch_size	No-epoch
Nagoya University	96,366	46,451 / 48%	10 (concat to 100)	5	256	Sigmoid	Gradient Descent	0.00006	128	50

Case study: test settings and results



- Number of tests:
  → 51,516
- Total accuracy range for tests:
  → 50.40 % - 53.46 %

Our initial approach (first model)

- **Assumptions**:
  1. Future trend consists of past information.
  2. We expect that case-study followed common naming when talking about layers.
     4+1 = 5 layer setting expected
  3. We assume (based on the presentation) they used 48% of data for training in initial case
  4. There is no ‘stall’ when price doesn’t move. We label it as Down.

- **Prediction types**: Classification: { Up, Down }

- **Dataset**:
  → USD/THB → 13/2/2017 ~ 13/10/2017 by Dukascopy online
  → At first 5833 instances, after removing 0-volume (noises) days: 3785 instances

First model: Data preprocessing

Raw data

	A	B	C	D	E	F
15	13.02.2017 13:00:0C	35.0603	35.0825	35.0397	35.0439	1212959961
16	13.02.2017 14:00:0C	35.0499	35.0917	35.036	35.0512	2651639893
17	13.02.2017 15:00:0C	35.0498	35.079	35.0298	35.0751	2516360107
18	13.02.2017 16:00:0C	35.0738	35.0831	35.0267	35.0467	3030510010
19	13.02.2017 17:00:0C	35.0589	35.0822	35.013	35.0389	2871860107
20	13.02.2017 18:00:0C	35.0486	35.0625	35.016	35.0485	1476300049
21	13.02.2017 19:00:0C	35.0479	35.0722	35.017	35.0557	1307040039

Processed data

	A	B	C	D	E	F	G	H	I	J	K
1	13	35.0603	35.0439	35.0825	35.0397	1212959961	35.05769444	0	9.813084112	95.60997363	1
2	14	35.0499	35.0512	35.0917	35.036	2651639893	35.05587222	32.4032974	27.28904847	94.31458809	1
3	15	35.0498	35.0751	35.079	35.0298	2516360107	35.05452778	68.44933179	73.1825525	93.76269537	0
4	16	35.0738	35.0467	35.0831	35.0267	3030510010	35.05474167	40.68560043	30.76923077	93.37176455	0
5	17	35.0589	35.0389	35.0822	35.013	2871860107	35.05738333	36.32742972	32.90978399	91.97953707	1
6	18	35.0486	35.0485	35.0625	35.016	1476300049	35.06072778	44.24367054	45.10800508	92.00689397	1
7	19	35.0479	35.0557	35.0722	35.017	1307040039	35.06427222	49.33169233	54.2566709	92.02741164	1

First model: Training settings

T1	Instances in dataset	Train / % train	Features	Layers	Neurons (total)
Nagoya University	96,366	46,451 / 48%	10 (concat to 100)	5	256
Assumption University	3785	1821 / 48%	10	5	256 in hidden layers

T2	Activation	Optimization	Learning-rate	Batch_size	No-epoch
Nagoya University	Sigmoid	Gradient Descent	0.00006	128	50
Assumption University	Sigmoid	Gradient Descent	0.00006	128	50

First model: test settings and outcomes

Test settings:

- 4 tests, each having 400 test instances and testing different parts of the dataset.

T3	# test instances	% accuracy
Nagoya University	744 - 51516	50.40 % - 53.46 %
Assumption University	400 - 1600	50.50 % - 54.75 %

First model: Conclusion

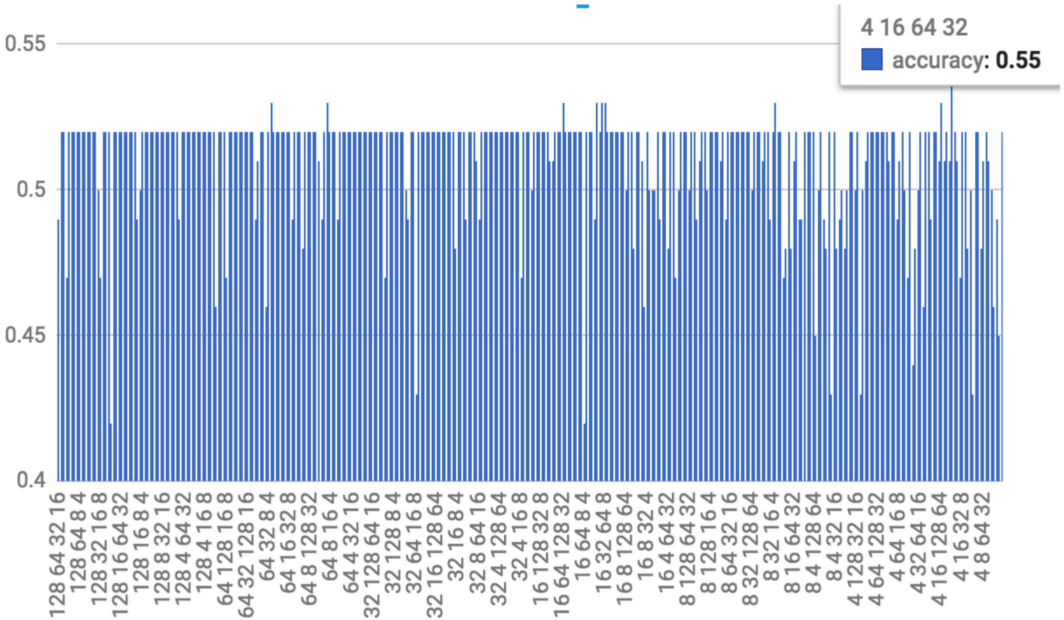
- Accuracy of our initial model and the case study are almost exactly alike
- Assumptions were not affecting negatively
- The intentional changes did not affect negatively (as expected)
- Even though successfully followed the case study’s results
  → **Not really a great level of accuracy**

Content

Development stage: neuron-layer setup (setting)

- Permutations (6,4) ⇒ 360 possible rounds
- Dataset ⇒ 3,785 instances
- Training set ⇒ 100 instances
- Num_Test ⇒ 100
- Optimizer ⇒ Gradient Descent
- Activation func. ⇒ ReLU
- Number of epoch ⇒ 50
- Batch size ⇒ 38
- Optimization steps ⇒ (100 / 38 * 50)= 198 steps

Development stage: neuron-layer setup (result)

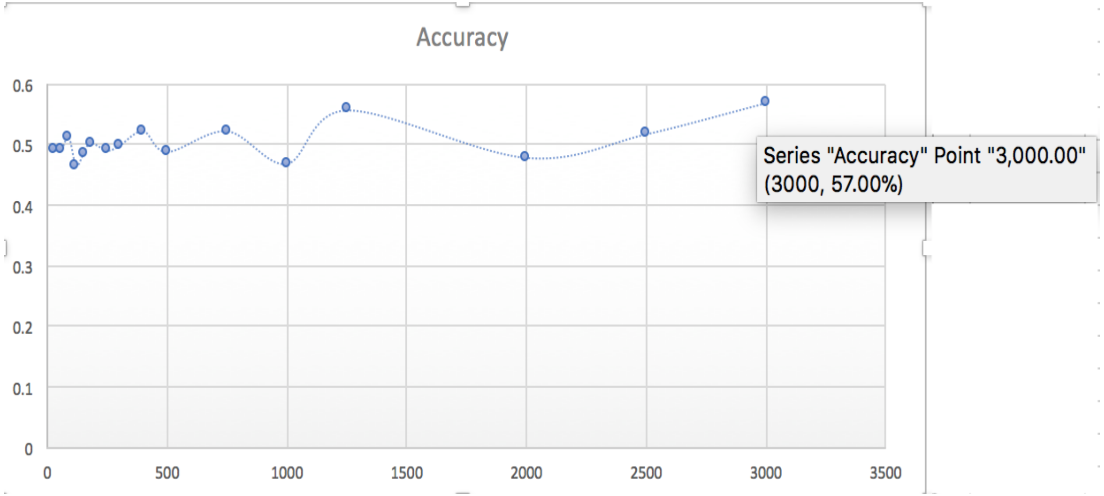


- Mean ⇒ 0.51
- Mode ⇒ 0.52
- Range ⇒ 0.42 ~ 0.55
- Best Result ⇒ { 4, 16, 64, 32}

Development stage: no. of training-instances (setting)

- Number of rounds ⇒ 17 rounds with each training instance
- Training instances :
    - [30, 60, 90, 120, 150, 180, 250, 300, 400, 500, 750, 1000, 1250, 1500, 2000, 2500, 3000]
- Dataset ⇒ 3,785 instances
- Testing set ⇒ 400 instances of sample size
- Optimizer ⇒ ProxmialAdagradOptmizer
- Activation func. ⇒ ReLU
- Learning_rate ⇒ 0.00006
- Number of epoch ⇒ 50
- Batch size ⇒ 128

Development stage: no. of training-instances (setting)



- Mean ⇒ 0.50
- Mode ⇒ 0.49 & 0.52
- Range ⇒ 0.46 ~ 0.57
- Best Result ⇒ *3,000 with 57% accuracy*

Development stage: intuition from the tests

Focus:

- Adjusting the named parameters don’t improve accuracy much
- Along with adjustment, optimal number of training instances becomes smaller
- Movement of the market affects on overall accuracy

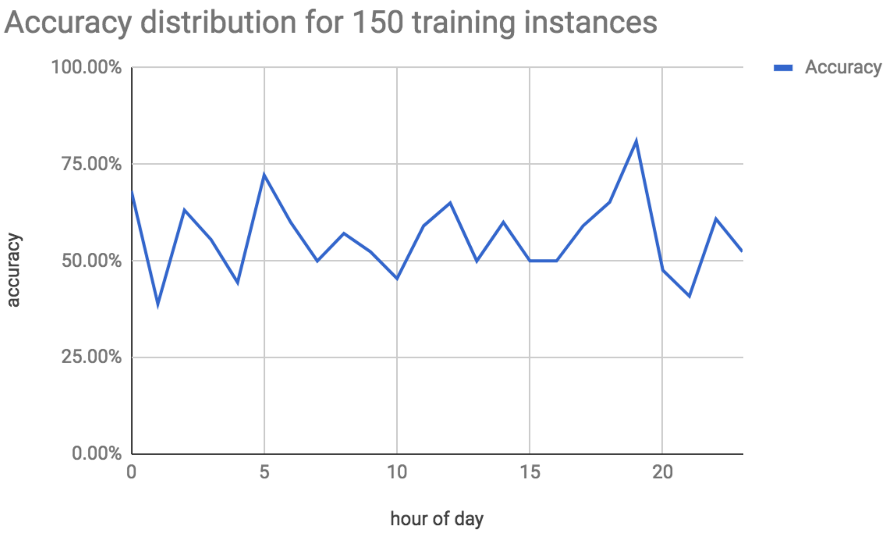
Intuition:

- Num_epoch ⇒ 250
- Learning rate ⇒ 0.0006
- Batch size ⇒ 38

Development stage: optimal prediction times

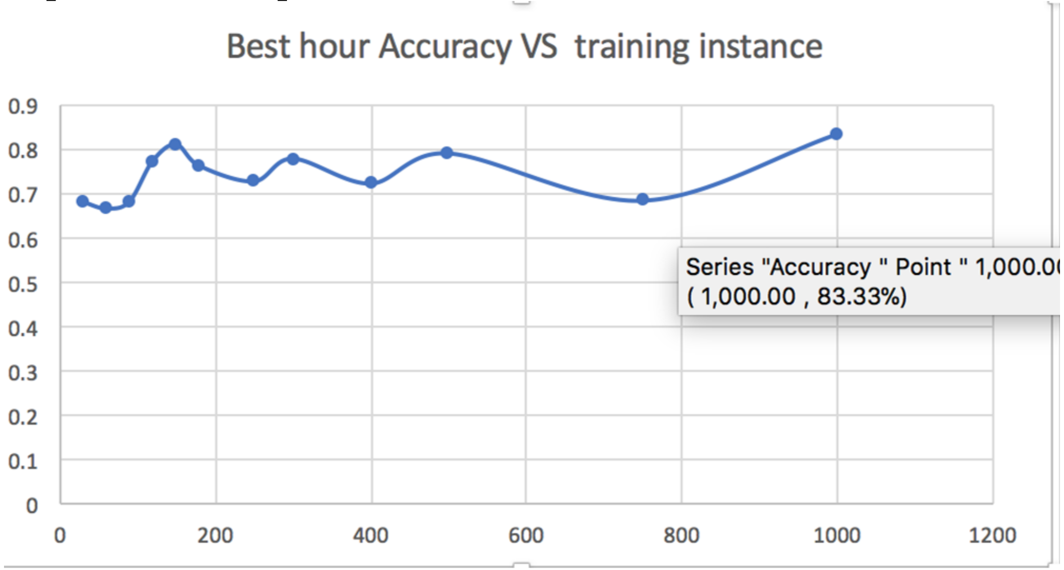
- Tested train-instance numbers: [30, 60, 90, 120, 150, 180, 250, 300, 400, 500, 750, 1000, 1250]
- Dataset ⇒ 3,785 instances
- Testing set ⇒ 500 tests (for each train-instance test)
- Optimizer ⇒ ProxmialAdagradOptmizer
- Activation func. ⇒ ReLU
- Learning_rate ⇒ 0.0006
- Number of epoch ⇒ 250
- Batch size ⇒ 38

optimal prediction times with 150 instances



Hours of the day	Accuracy	Hours of the day	Accuracy
0	68.18%	12	65.00%
1	38.89%	13	50.00%
2	63.16%	14	60.00%
3	55.56%	15	50.00%
4	44.44%	16	50.00%
5	72.22%	17	59.09%
6	60.00%	18	65.22%
7	50.00%	19	80.95%
8	57.14%	20	47.62%
9	52.38%	21	40.91%
10	45.45%	22	60.87%
11	59.09%	23	52.38%

Development stage: optimal prediction times



Instances	Hour	Accuracy
30.00	13	68.18%
60.00	8	66.67%
90.00	17	68.18%
120.00	16	77.27%
150.00	19	**80.95%**
180.00	8	76.19%
250.00	11	72.73%
300.00	4	77.78%
400.00	5	72.22%
500.00	2	78.95%
750.00	2	68.42%
1,000.00	23	**83.33%**

Content

First ANN-learning case

- Very first model based on case study and other assumptions
- Results obtained : range of 50-54 %
- Able to obtain exactly same range of accuracy as the case study
- Result range was as expected as the case study provided result
- Provided us a good foundations for deeper level experiments for future testings

Goals

CHECKLIST:

- ✅ Comprehend and apply the Case-study's approach in the Initial design
- ☐ Branch-off and find better models using our own techniques
- ☐ Apply the model in a tangible application use-case

Development stages

1. Finding optimal neuron-layer setup ⇒ { 4, 16, 64, 32} with 55.00%
2. Optimizing training-instance settings ⇒ *3,000 training instances with 57% accuracy*
3. Intuition of the tests ⇒ Num_epoch : 250 , Learning rate : 0.0006 , Batch size : 38
4. Optimal prediction times ⇒ {150 , 19th hr , 80.95%} & {1,000 , 23rd hr , 83.33%}

Goals

CHECKLIST:

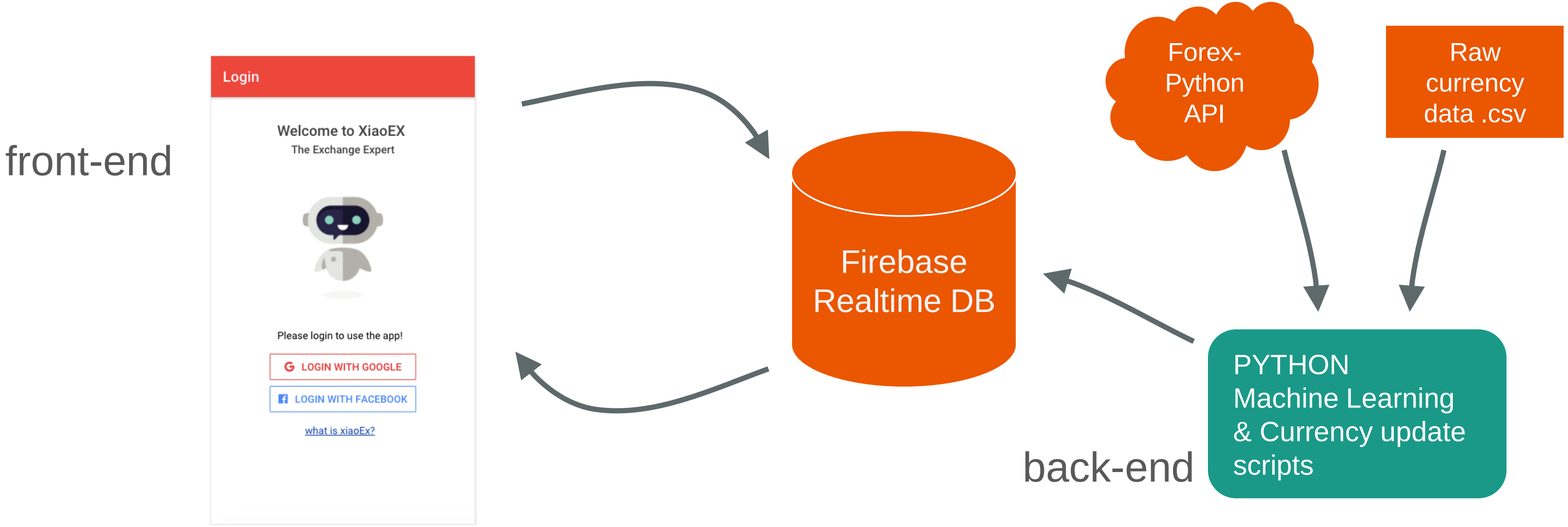
- ✅ Comprehend and apply the Case-study’s approach in the Initial design
- ✅ Branch-off and find better models using our own techniques
- ☐ Apply the model in a tangible application use-case

Content

Application use-case

- Python back-end service
- A hybrid mobile app serve for currency predictions and exchange rate

Architecture of the App use-case



Application Demo

https://xiaoexau.firebaseapp.com/

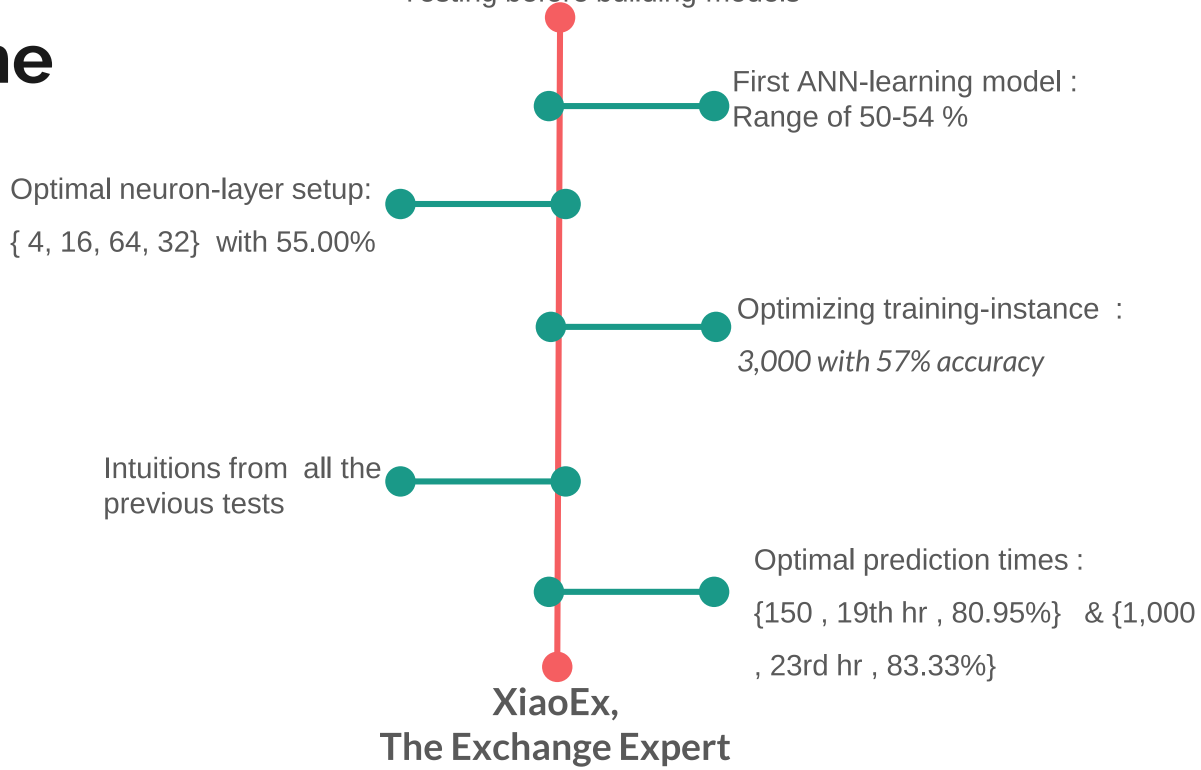
Goals

CHECKLIST:

- ✅ Comprehend and apply the Case-study’s approach in the Initial design
- ✅ Branch-off and find better models using our own techniques
- ✅ Apply the model in a tangible application use-case

XiaoEx's timeline

- Case study
- Neural Network & Readings
- Data processing
- Testing before building models



Conclusion

This project has provided us a very interesting insight of the possibilities of Machine Learning

To be the first machine learning task we think we succeeded in the task given the complexity of the given problem and background knowledge of the members.

We feel this project serves as an excellent foundation to dig deeper into the field of Machine Learning

Questions & Answers