NIO

Exercise 09: MLP

Duc Duy Pham, M.Sc.

Raum: BC 410
Tel.: 0203-379-3734
Email: duc.duy.pham@uni-due.de

Content

- Revision Practical Tasks (Already uploaded)
- Revision Lecture (MLP)
- New Task (Already uploaded)

Content

- Revision Practical Tasks (Already uploaded)
- **Revision Lecture (MLP)**
- New Task (Already uploaded)

Revision

- What are the trainable parameters of a MLP?

Revision

- What are the trainable parameters of a MLP?
  1) Number of layers
  2) Connection weights between layers
  3) Number of neurons per layer
  4) Bias for each neuron

UNIVERSITÄT DUISBURG ESSEN IS

Offen im Denken

Revision

- What are the trainable parameters of a MLP?
  1) Number of layers
  2) Connection weights between layers
  3) Number of neurons per layer
  4) Bias for each neuron

A: 1, 3

B: 2, 4

C: 2, 3

D: 1, 4

Revision

- What are the trainable parameters of a MLP?
  1) Number of layers
  2) Connection weights between layers
  3) Number of neurons per layer
  4) Bias for each neuron

A: 1, 3

B: 2, 4

C: 2, 3

D: 1, 4

Revision

- Given a 3-Layer MLP with K input neurons and L, M, N neurons (units) in the other layers respectively, how many trainable parameters, does this MLP have?

Revision

- Given a 3-Layer MLP with K input neurons and L, M, N neurons (units) in the other layers respectively, how many trainable parameters, does this MLP have?
  1) $(K+1)*L + (L+1)*M + (M+1)*N$
  2) $K*L + L*M + M*N$
  3) $K*(L+1) + L*(M+1) +M*(N+1)$
  4) $K*L*M*N$

Offen im Denken

Revision

- Given a 3-Layer MLP with K input neurons and L, M, N neurons (units) in the other layers respectively, how many trainable parameters, does this MLP have?
  1) (K+1)*L + (L+1)*M + (M+1)*N
  2) K*L + L*M + M*N
  3) K*(L+1) + L*(M+1) +M*(N+1)
  4) K*L*M*N

A: 1

B: 2

C: 3

D: 4

Offen im Denken

Revision

- Given a 3-Layer MLP with K input neurons and L, M, N neurons (units) in the other layers respectively, how many trainable parameters, does this MLP have?

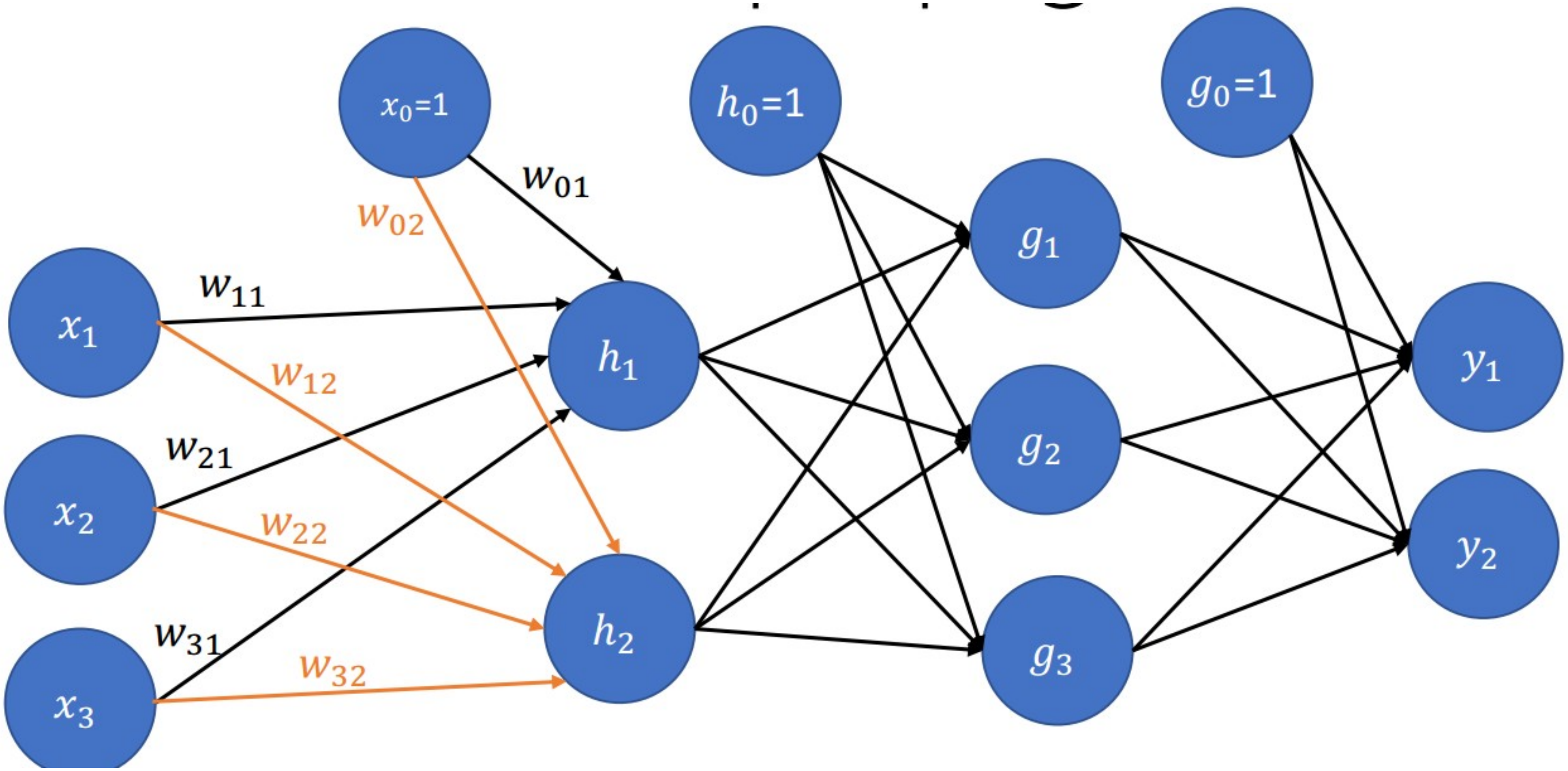
1) $(K+1)*L + (L+1)*M + (M+1)*N$
2) $K*L + L*M + M*N$
3) $K*(L+1) + L*(M+1) + M*(N+1)$
4) $K*L*M*N$

A: 1

B: 2

C: 3

D: 4


x_0=1
h_0=1
g_0=1
w_{01}
w_{02}
w_{11}
w_{12}
w_{21}
w_{22}
w_{31}
w_{32}
x_1
x_2
x_3
h_1
h_2
g_1
g_2
g_3
y_1
y_2

Revision of lecture

- What statements regarding Backpropagation are true?

Revision of lecture

- What statements regarding Backpropagation are true?
  1) Backpropagation is the same as gradient descent
  2) Backpropagation is used to establish gradient descent
  3) Backpropagation is a universal optimization method
  4) Backpropagation is used to update the weights

Revision of lecture

- What statements regarding Backpropagation are true?
  1) Backpropagation is the same as gradient descent
  2) Backpropagation is used to establish gradient descent
  3) Backpropagation is a universal optimization method
  4) Backpropagation is used to update the weights

A: 1, 3

B: 2, 4

C: 2, 3

D: 1, 4

Revision of lecture

- What statements regarding Backpropagation are true?
  1) Backpropagation is the same as gradient descent
  2) Backpropagation is used to establish gradient descent
  3) Backpropagation is a universal optimization method
  4) Backpropagation is used to update the weights

A: 1, 3

B: 2, 4

C: 2, 3

D: 1, 4

Revision of lecture

- What are the 3 main steps for Backpropagation Learning?

Revision of lecture

- What are the 3 main steps for Backpropagation Learning?
  - Feed Forward
  - Estimating Gradient through Backpropagation
  - Weight update

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Revision of lecture

- What does the following assignment (from the lecture) stand for?

$$w_{l,i,j}^{t+1} := w_{l,i,j}^{t} - \mu \left.\frac{\partial D(w)}{\partial w_{l,i,j}}\right|_{w_{l,i,j}^{t}}$$

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Revision of lecture

- What does the following assignment (from the lecture) stand for?

$$w_{l,i,j}^{t+1} := w_{l,i,j}^{t} - \mu \left. \frac{\partial D(w)}{\partial w_{l,i,j}} \right|_{w_{l,i,j}^{t}}$$

1) Weight update by Backpropagation
2) Weight update by Gradient Descent
3) Perceptron Learning Rule
4) Proportional Learning Rule

Revision of lecture

- What does the following assignment (from the lecture) stand for?

$$w_{l,i,j}^{t+1} := w_{l,i,j}^{t} - \mu \left.\frac{\partial D(w)}{\partial w_{l,i,j}}\right|_{w_{l,i,j}^{t}}$$

1) Weight update by Backpropagation
2) Weight update by Gradient Descent
3) Perceptron Learning Rule
4) Proportional Learning Rule

A: 1

B: 2

C: 3

D: 4

Revision of lecture

- What does the following assignment (from the lecture) stand for?

$$w_{l,i,j}^{t+1} := w_{l,i,j}^{t} - \mu \left. \frac{\partial D(w)}{\partial w_{l,i,j}} \right|_{w_{l,i,j}^{t}}$$

1) Weight update by Backpropagation
2) Weight update by Gradient Descent
3) Perceptron Learning Rule
4) Proportional Learning Rule

A: 1

B: 2

C: 3

D: 4

Revision of lecture

- Why not only propagate „forward“?

Revision of lecture

- Why not only propagate „forward“?
  1) For calculation of error, error of next layer is required first
  2) For calculation of partial derivative of specific weight, partial derivatives of next layer are required first
  3) Gradient Descent always demands backpropagation
  4) It is the same

Revision of lecture

- Why not only propagate „forward“?
  1) For calculation of error, error of next layer is required first
  2) For calculation of partial derivative of specific weight, partial derivatives of next layer are required first
  3) Gradient Descent always demands backpropagation
  4) It is the same

A: 1,2,3

B: 2,3

C: 2,4

D: 2

Revision of lecture

- Why not only propagate „forward“?
  1) For calculation of error, error of next layer is required first
  2) For calculation of partial derivative of specific weight, partial derivatives of next layer are required first
  3) Gradient Descent always demands backpropagation
  4) It is the same

A: 1,2,3

B: 2,3

C: 2,4

D: 2

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \boxed{\frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}} \longrightarrow \boxed{\text{Lecture}}$$

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \boxed{\frac{\partial d^m(w)}{\partial y_{\ell,j}}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\boxed{\frac{\partial d^m(w)}{\partial y_{\ell,j}}}$$

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\frac{\partial d^m(w)}{\partial y_{\ell,j}} = \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \frac{\partial y_{\ell+1,n}}{\partial y_{\ell,j}}$$

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\frac{\partial d^m(w)}{\partial y_{\ell,j}} = \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \frac{\partial y_{\ell+1,n}}{\partial y_{\ell,j}}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \quad \frac{\partial}{\partial y_{\ell,j}} \left[f_\sigma \left(\sum_{q=0}^{N_\ell} w_{\ell+1,q,n} \cdot y_{\ell,q} \right) \right]$$

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\frac{\partial d^m(w)}{\partial y_{\ell,j}} = \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \frac{\partial y_{\ell+1,n}}{\partial y_{\ell,j}}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \quad \frac{\partial}{\partial y_{\ell,j}} \left[f_\sigma \left(\sum_{q=0}^{N_\ell} w_{\ell+1,q,n} \cdot y_{\ell,q} \right) \right]$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} f'_\sigma(u_{\ell+1,n}) \frac{\partial}{\partial y_{\ell,j}} u_{\ell+1,n}$$

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\frac{\partial d^m(w)}{\partial y_{\ell,j}} = \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \frac{\partial y_{\ell+1,n}}{\partial y_{\ell,j}}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \quad \frac{\partial}{\partial y_{\ell,j}} \left[f_\sigma \left(\sum_{q=0}^{N_\ell} w_{\ell+1,q,n} \cdot y_{\ell,q} \right) \right]$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} f'_\sigma(u_{\ell+1,n}) \frac{\partial}{\partial y_{\ell,j}} u_{\ell+1,n}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} y_{\ell+1,n}(1 - y_{\ell+1,n}) \cdot w_{\ell+1,j,n}$$

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\frac{\partial d^m(w)}{\partial y_{\ell,j}} = \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \frac{\partial y_{\ell+1,n}}{\partial y_{\ell,j}}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \quad \frac{\partial}{\partial y_{\ell,j}} \left[f_\sigma \left(\sum_{q=0}^{N_\ell} w_{\ell+1,q,n} \cdot y_{\ell,q} \right) \right]$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \boxed{f'_\sigma(u_{\ell+1,n})} \frac{\partial}{\partial y_{\ell,j}} u_{\ell+1,n}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \boxed{y_{\ell+1,n}(1 - y_{\ell+1,n})} \cdot w_{\ell+1,j,n}$$

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\frac{\partial d^m(w)}{\partial y_{\ell,j}} = \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \frac{\partial y_{\ell+1,n}}{\partial y_{\ell,j}}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \quad \frac{\partial}{\partial y_{\ell,j}} \left[f_\sigma \left(\sum_{q=0}^{N_\ell} w_{\ell+1,q,n} \cdot y_{\ell,q} \right) \right]$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} f'_\sigma(u_{\ell+1,n}) \boxed{\frac{\partial}{\partial y_{\ell,j}} u_{\ell+1,n}}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} y_{\ell+1,n}(1 - y_{\ell+1,n}) \cdot \boxed{w_{\ell+1,j,n}}$$

UNIVERSITÄT DUISBURG ESSEN

IS

Offen im Denken

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\frac{\partial d^m(w)}{\partial y_{\ell,j}} = \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \frac{\partial y_{\ell+1,n}}{\partial y_{\ell,j}}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \frac{\partial}{\partial y_{\ell,j}} \left[f_\sigma \left(\sum_{q=0}^{N_\ell} w_{\ell+1,q,n} \cdot y_{\ell,q} \right) \right]$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} f'_\sigma(u_{\ell+1,n}) \boxed{\frac{\partial}{\partial y_{\ell,j}} u_{\ell+1,n}}$$

$$\boxed{\frac{\partial}{\partial y_{\ell,j}} \left[\quad \left(\sum_{q=0}^{N_\ell} w_{\ell+1,q,n} \cdot y_{\ell,q} \right) \right]}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} y_{\ell+1,n}(1 - y_{\ell+1,n}) \cdot \boxed{w_{\ell+1,j,n}}$$

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\frac{\partial d^m(w)}{\partial y_{\ell,j}} = \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \frac{\partial y_{\ell+1,n}}{\partial y_{\ell,j}}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \quad \frac{\partial}{\partial y_{\ell,j}} \left[f_\sigma \left(\sum_{q=0}^{N_\ell} w_{\ell+1,q,n} \cdot y_{\ell,q} \right) \right]$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} f'_\sigma(u_{\ell+1,n}) \frac{\partial}{\partial y_{\ell,j}} u_{\ell+1,n}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} y_{\ell+1,n}(1 - y_{\ell+1,n}) \cdot w_{\ell+1,j,n}$$

$$\frac{\partial d^m(w)}{\partial w_{\ell,i,j}} = \frac{\partial d^m(w)}{\partial y_{\ell,j}} \cdot \frac{\partial y_{\ell,j}}{\partial w_{\ell,i,j}}$$

$$\boxed{\frac{\partial d^m(w)}{\partial y_{\ell,j}}} = \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \frac{\partial y_{\ell+1,n}}{\partial y_{\ell,j}}$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} \cdot \quad \frac{\partial}{\partial y_{\ell,j}} \left[f_\sigma \left(\sum_{q=0}^{N_\ell} w_{\ell+1,q,n} \cdot y_{\ell,q} \right) \right]$$

$$= \sum_{n=1}^{N_{\ell+1}} \frac{\partial d^m(w)}{\partial y_{\ell+1,n}} f'_\sigma(u_{\ell+1,n}) \frac{\partial}{\partial y_{\ell,j}} u_{\ell+1,n}$$

$$= \sum_{n=1}^{N_{\ell+1}} \boxed{\frac{\partial d^m(w)}{\partial y_{\ell+1,n}}} y_{\ell+1,n}(1 - y_{\ell+1,n}) \cdot w_{\ell+1,j,n}$$

Training MLPs

- What is Batch-Learning?

Training MLPs

- What is Batch-Learning?
  - Feed the network multiple samples
  - Aggregate loss over these number of samples (batch size)
  - Weight update not for single sample but for batch of samples!
  - Weight fluctuation is reduced

Training MLPs

- What are epochs?

Training MLPs

- What are epochs?
  - Once the whole training data set is used up, shuffle training data set and use it again
  - Number of times the whole training data set is re-used = number of epochs
  - Caution:
    This might lead to overfitting to the training data set!

Training MLPs

- How can we prevent overfitting/choose hyperparameters?

UNIVERSITÄT DUISBURG ESSEN

IS

Training MLPs

- How can we prevent overfitting/choose hyperparameters?
  - Separate data into training, validation and test data!
  - Training data to train your model (even multiple epochs)
  - Validation data for hyper parameter tuning (e.g. early stopping to prevent overfitting)
  - Test data to evaluate your model (= assumed real world data)

Training MLPs

- MLP topology for classification task?

Training MLPs

- MLP topology for classification task?
  - Recommended final layer:
    - Number of output neurons = number of classes (one hot coding)
    - Softmax as activation function

$$p_i := \frac{\exp(f_{p\,i})}{\sum_k \exp(f_{p\,k})}$$

Training MLPs

- MLP topology for classification task?
  - Recommended final layer:
    - Number of output neurons = number of classes (one hot coding)
    - Softmax as activation function

$$p_i := \frac{\exp(f_{p\,i})}{\sum_k \exp(f_{p\,k})}$$

  - Recommended loss function:
    - Cross Entropy:

$$-\sum_i y_i \log(p_i)$$

Content

- Revision Practical Tasks (Already uploaded)
- Revision Lecture (MLP)
- **New Task (Already uploaded)**

Jupyter Notebook

- Implement a Multi Layer Perceptron in tensorflow