Deep Learning

Exercise 5: Binary and Multi-class Classification

Instructor: Manuel Günther
Email: guenther@ifi.uzh.ch
Office: AND 2.54

Friday, March 26, 2020

Outline

Outline

Dataset

Classification of Forged Bank Notes

- Download dataset from UCI
- Inputs: 4 attributes
  1. variance of Wavelet Transformed image
  2. skewness of Wavelet Transformed image
  3. curtosis of Wavelet Transformed image
  4. entropy of image
- Output: Forged (0) or real bank note (1)
- Task: how many hidden neurons to reach 100% accuracy?

Dataset URL

http://archive.ics.uci.edu/ml/datasets/banknote+authentication

Dataset

Classification of Spam Emails

- Download dataset from UCI
- Inputs: 58 attributes
  - → 48 relative number of occurrences of specific key words
  - → 6 relative number of occurrences of specific key characters
  - → 3 counts according to capital letters
- Output: Spam (1) or not spam (0)
- Task: get highest classification accuracy

Dataset URL

http://archive.ics.uci.edu/ml/datasets/Spambase

Binary Classification

Task 1: Load Dataset

- Read line by line and split
- First values: $\vec{x}$
- Last value: t
- Turn to $\mathbf{X}$ and $\vec{t}$
  - $\rightarrow$ Remember to add x_0

Task 2: Implement Network

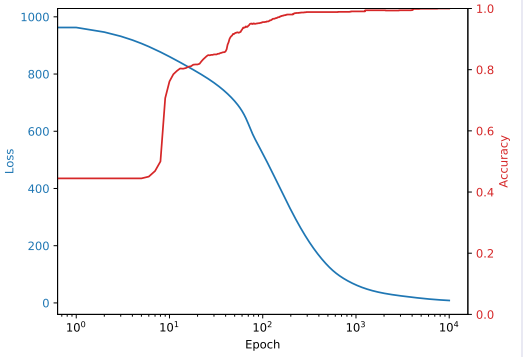
- `def forward(X, W1, W2):`
- Hidden unit output $\mathbf{H}$
- Logits $\vec{z}$
- Network output $\vec{y} = \sigma(\vec{z})$

Task 3: Gradient Descent

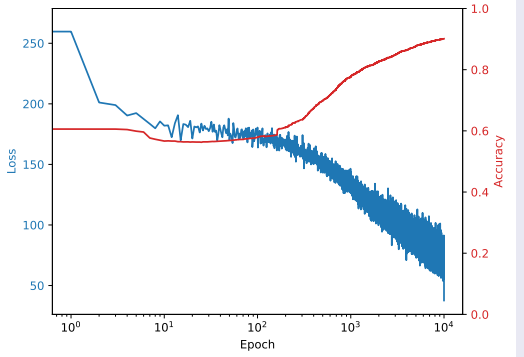
- `def descent(X, T, W1, W2, eta):`
- Forward pass: $\vec{y} = f(\mathbf{X})$
- Compute loss:
  $\mathcal{J}^{\mathrm{CE}} = \vec{t}^{\mathrm{T}} \ln \vec{y} + (1 - \vec{t})^{\mathrm{T}} \ln(1 - \vec{y})$
- Compute gradients: $\nabla_{\mathbf{W}^{(1)}}, \nabla_{\vec{w}^{(2)}}$
  - $\rightarrow$ Same as last week
- Perform weight update:
  $\mathbf{W}^{(1)} -= \eta \nabla_{\mathbf{W}^{(1)}} \quad \vec{w}^{(2)} -= \eta \nabla_{\vec{w}^{(2)}}$
- Train network for 10000 epochs

Example Losses

Banknote Forgery



Spam Classification



Outline

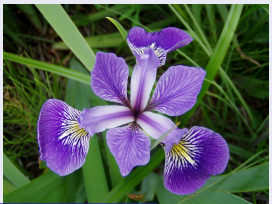
The Iris Flower Dataset

Description

- 4 input variables:
  1. Sepal length in cm
  2. Sepal width in cm
  3. Petal length in cm
  4. Petal width in cm
- 3 different classes:
  1. Iris Setosa
  2. Iris Versicolor
  3. Iris Virginica
- Available in `sklearn`
- Task: 100% accuracy

Setosa

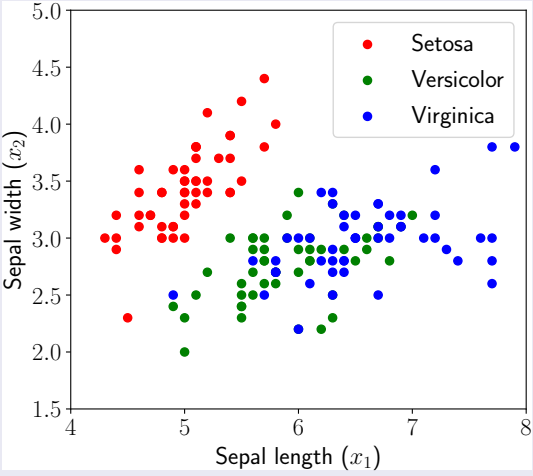
Versicolor



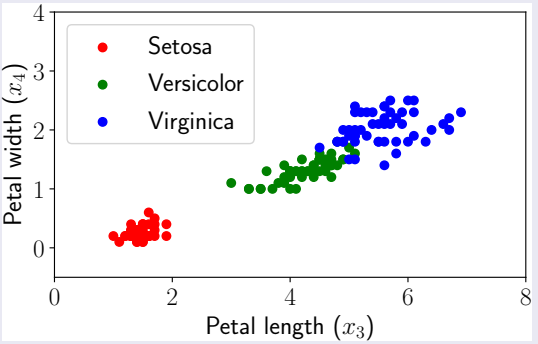
Virginica

The Iris Flower Dataset

Distribution in (x_1, x_2)



Distribution in (x_3, x_4)

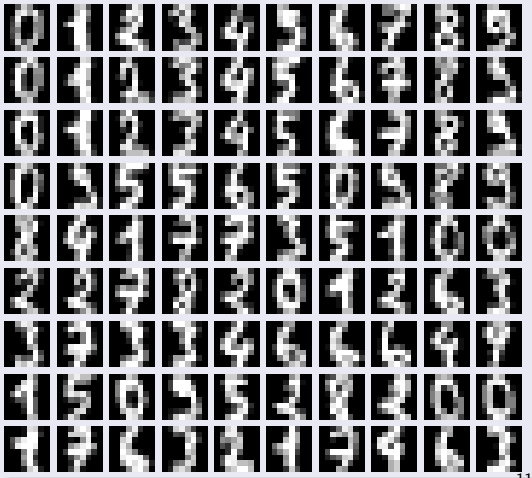


The Digits Dataset

Description

- 64 input variables:
- Pixel values of 8×8 images
  - $\rightarrow$ Value range $[0, 16]$
  - $\rightarrow$ Given as 64-element 1D array
- 10 different classes (digit 0-9)
- Available in `sklearn`
- Task: highest accuracy

Example Images



Categorical Classification

Task 1: Load Dataset

- Load data from sklearn
- Turn to $\mathbf{X}$ and $\mathbf{T}$
  - $\rightarrow$ Remember to add x_0

Task 2: Implement Network

- `def forward(X, W1, W2):`
- Hidden unit output $\mathbf{H}$
- Logits $\mathbf{Z}$
- Network output $\mathbf{Y} = S(\mathbf{Z})$

Task 3: Compute Accuracy

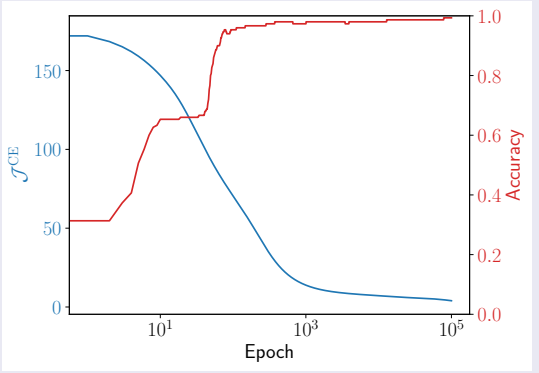
- `def accuracy(X, T, W1, W2):`

Task 3: Gradient Descent Step

- `def descent(X, T, W1, W2, eta):`
- Forward pass: $\vec{y} = f(\mathbf{X})$
- Compute loss:
$$\mathcal{J}^{\mathrm{CE}} = -\sum_{n=1}^{N}\left[z_{t^{[n]}}^{[n]} - \ln \sum_{o=1}^{O} e^{z_o^{[n]}}\right]$$
- Compute gradients: $\nabla_{\mathbf{W}^{(1)}}, \nabla_{\mathbf{W}^{(2)}}$
  - $\rightarrow$ Same as last week
- Perform weight update:
$$\mathbf{W}^{(1)} -= \eta\nabla_{\mathbf{W}^{(1)}} \quad \mathbf{W}^{(2)} -= \eta\nabla_{\mathbf{W}^{(2)}}$$
- Train network for 10000 epochs

Example Losses

Iris Dataset



Digits Dataset