Deep Learning

Übung WS 23/24

Judith Nester (nester@uni-koeln.de)

07-12-2023

Recap

Feed-Forward Neural Networks

- Neurons with logistic regressions in them
- Layered architecture: Multiple neurons in parallel
- Forward pass: Predict with a trained network
- Back propagation: Train a network (similar to gradient descent)
- Implementation in keras

```
ffnn = Sequential()
ffnn.add(Input(shape=(3,)))
ffnn.add(Dense(5, activation="sigmoid"))
ffnn.add(Dense(1, activation="softmax"))

ffnn.compile(loss="mean_squared_error", optimizer="sgd")

ffnn.fit(train_data, train_labels, epochs=150, batch_size=2)
```

- Exercise 6

Today

Representing Text (Part 1)

Overfitting

Regularization

Dropout

Exercise

Section 1

Representing Text (Part 1)

Introduction

» So far: Explicitly defined features
- Still relevant for all kinds of structured data (e.g., tabular material)

Introduction

- So far: Explicitly defined features
  - Still relevant for all kinds of structured data (e.g., tabular material)
- Text classification: What features to use?
  - NLP, until ∼2013: Finding, developing and implementing features as a scientific task
  - NLP, today: Bag of words
    - . . . and embeddings, next week

Bag of Words

- Disregard word order
- Count number of words
- Frequency of the words → vector that represents the text

Bag of Words (BOW)

Example

Review: »The best soundtrack ever to anything.: I'm reading a lot of reviews saying that this is the best 'game soundtrack' and I figured that I'd write a review to disagree a bit. This in my opinino is Yasunori Mitsuda's ultimate masterpiece. The music is timeless and I'm been listening to it for years now and its beauty simply refuses to fade.The price tag on this is pretty staggering I must say, but if you are going to buy any cd for this much money, this is the only one that I feel would be worth every penny.«

Bag of Words (BOW)

Example

Review: »The best soundtrack ever to anything.: I'm reading a lot of reviews saying that this is the best 'game soundtrack' and I figured that I'd write a review to disagree a bit. This in my opinino is Yasunori Mitsuda's ultimate masterpiece. The music is timeless and I'm been listening to it for years now and its beauty simply refuses to fade.The price tag on this is pretty staggering I must say, but if you are going to buy any cd for this much money, this is the only one that I feel would be worth every penny.«

Bag of words: 0, 0, 3, 1, 0, 0, 1, 0, 1, 0, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 1, 5, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 3, 5, 0, 5, 0, 5, 0, 0, 0, 0, 0, 1, 1

Bag of Words (BOW)

Example

Review: »The best soundtrack ever to anything.: I'm reading a lot of reviews saying that this is the best 'game soundtrack' and I figured that I'd write a review to disagree a bit. This in my opinino is Yasunori Mitsuda's ultimate masterpiece. The music is timeless and I'm been listening to it for years now and its beauty simply refuses to fade.The price tag on this is pretty staggering I must say, but if you are going to buy any cd for this much money, this is the only one that I feel would be worth every penny.«

Bag of words: 0, 0, 3, 1, 0, 0, 1, 0, 1, 0, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 1, 5, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 3, 5, 0, 5, 0, 5, 0, 0, 0, 0, 0, 1, 1

Bag of Words (BOW)

Noteworthy

» Limited vocabulary: In reality, one cannot include all words
- Vocabulary size depends on corpus size
- Most frequent tokens

Bag of Words (BOW)

Noteworthy

- Limited vocabulary: In reality, one cannot include all words
  - Vocabulary size depends on corpus size
  - Most frequent tokens
- Fitting
  - Generating a BOW representation requires two passes over data
    - `fit(...)` and `transform(...)`

Bag of Words (BOW)

Implementation

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(
max_features=1000, # how many tokens to distinguish?
lowercase=True # make everything lower-cased?
)
# 'learn' mapping of words to dimensions
vectorizer.fit(texts)

# apply mapping to text
texts_vec = vectorizer.transform(texts)
```

Section 2

Overfitting

Introduction

» ›Fitting‹: Train a model on data (= »fit« it to the data)
- Underfitting: The model is not well fitted to the data, i.e., accuracy is low
- Overfitting: The model is fitted too well to the data, i.e., accuracy is high

Introduction

» ›Fitting‹: Train a model on data (= »fit« it to the data)

- Underfitting: The model is not well fitted to the data, i.e., accuracy is low
- Overfitting: The model is fitted too well to the data, i.e., accuracy is high

Why is overfitting a problem?

Introduction

» ›Fitting‹: Train a model on data (= »fit« it to the data)
- Underfitting: The model is not well fitted to the data, i.e., accuracy is low
- Overfitting: The model is fitted too well to the data, i.e., accuracy is high

Why is overfitting a problem?

» We want to the model to behave well »in the wild«

» It needs to generalize from training data

» If it is overfitted, it works very well on training data, and very badly on test data

Intuition

$\simeq$ Learning by heart

Example

» Learning by heart gets you through the test
- I.e., systems achieve high performance

Intuition

≃ Learning by heart

Example

- Learning by heart gets you through the test
  - I.e., systems achieve high performance
- You are unable to apply your knowledge to situations not exactly as in the test
  - I.e., system performance is lower in the wild

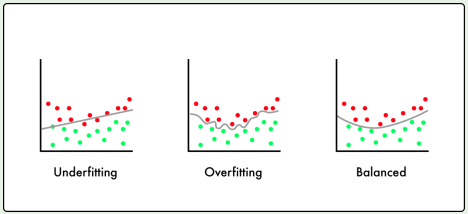

Figure: Towards Data Science

Overfitting and Neural Networks

Classical machine learning

- » Feature selection can avoid relying on irrelevant features
- ⚠ Only one source for overfitting

Overfitting and Neural Networks

Classical machine learning

» Feature selection can avoid relying on irrelevant features

⚠ Only one source for overfitting

Neural networks are overfitting machines

» Layered architecture $\Rightarrow$ Any relation between x and y can be learned

- including a fixed set of if/else rules

Techniques against overfitting

» Regularization

» Dropout

Real-World Examples

This is an excellent collection of examples for overfitting: `https://stats.stackexchange.com/questions/128616/whats-a-real-world-example-of-overfitting`

Section 3

Regularization

Formalization

» Formally, regularization is a parameter added to the loss

$$J(\vec{w}) = J_{\text{original}}(\vec{w}) + R$$

L^2-Regularization

L^2-Norm (a. k. a. Euclidean norm, ridge regression, weight decay)

- Tikhonov (1963)
- Given a vector $\vec{x} = (x_1, x_2, \ldots, x_n)$,
  its L^2 norm is $L^2(\vec{x}) = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = ||\vec{x}||_2$

L^2-Regularization

L^2-Norm (a. k. a. Euclidean norm, ridge regression, weight decay)

- Tikhonov (1963)
- Given a vector $\vec{x} = (x_1, x_2, \ldots, x_n)$,
  its L^2 norm is $L^2(\vec{x}) = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = ||\vec{x}||_2$
- In practice, we drop the square root and calculate L^2 norm of the weight vector during training:

$$(||\vec{w}||_2)^2 = \sum_{i=0}^{n} w_i^2$$

L^2-Regularization

L^2-Norm (a. k. a. Euclidean norm, ridge regression, weight decay)

- Tikhonov (1963)
- Given a vector $\vec{x} = (x_1, x_2, \ldots, x_n)$,
  its L^2 norm is $L^2(\vec{x}) = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = ||\vec{x}||_2$
- In practice, we drop the square root and calculate L^2 norm of the weight vector during training:

$$(||\vec{w}||_2)^2 = \sum_{i=0}^{n} w_i^2$$

- Regularization rate λ: Factor that expresses how much we want (another hyperparameter)
  - too weak: overfitting possible
  - too strong: model underestimates the weight –> underfitting possible

$$J(\vec{w}) = J_{\text{original}}(\vec{w}) + \frac{\lambda}{n}||w||_2^2 \qquad \text{with } n \text{ for the batch size}$$

L_2-Regularization

» What does it do?

L_2-Regularization

» What does it do?

- If weights $\vec{w}$ are large: Loss is increased more
- Large weights are only considered if the increased loss is »worth it«, i.e., if it is counterbalanced by a real error reduction
- Small weights are preferred

L^1-Regularization (Tibshirani 1996)

» Absolute values instead of squares

$$L^1(\vec{x}) = \sum_{i=0}^{n} |x_i|$$

L^1-Regularization (Tibshirani 1996)

» Absolute values instead of squares

$$L^1(\vec{x}) = \sum_{i=0}^{n} |x_i|$$

L^1 or L^2?

» Skansi 2018:

- In most cases: L^2 is better
- Use L^1 if data is very noisy or sparse

Implementation

» In Keras, most layers support additional arguments for regularization:

- `kernel_regularizer`, `bias_regularizer`, `activity_regularizer`
  - Applied to weights, constant term, neuron output (= result of activation function)
  - Docs: `https://keras.io/api/layers/regularizers/`

Implementation

» In Keras, most layers support additional arguments for regularization:

- `kernel_regularizer`, `bias_regularizer`, `activity_regularizer`
  - Applied to weights, constant term, neuron output (= result of activation function)
  - Docs: `https://keras.io/api/layers/regularizers/`
- Argument value: Regularization function with parameter(s)
  - Layer-specific

Implementation

» In Keras, most layers support additional arguments for regularization:

- `kernel_regularizer`, `bias_regularizer`, `activity_regularizer`
  - Applied to weights, constant term, neuron output (= result of activation function)
  - Docs: `https://keras.io/api/layers/regularizers/`
- Argument value: Regularization function with parameter(s)
  - Layer-specific

```
ffnn.add(layers.Dense(5,
activation="sigmoid",
activity_regularizer=regularizers.L2(0.2)))
```

Section 4

Dropout

Dropout

- Regularization: Numerically combatting overfitting
- Dropout: Structurally combatting overfitting
  - Hinton et al. (2012)

Dropout

- Regularization: Numerically combatting overfitting
- Dropout: Structurally combatting overfitting
  - Hinton et al. (2012)
  - A new hyperparameter $\pi = [0; 1]$
  - In each epoch, every weight is set to zero with a probability of π

Dropout

Example

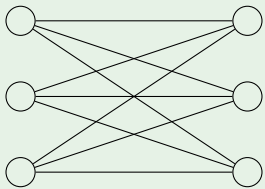

Figure: Dropout $\pi = 0.5$, visualized

Dropout

Example

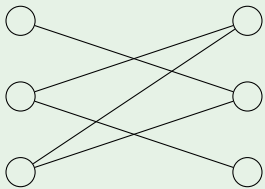

Figure: Dropout $\pi = 0.5$, visualized, Epoch 0

Dropout

Example

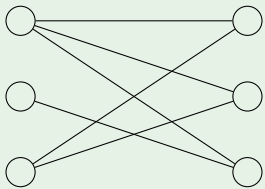

Figure: Dropout $\pi = 0.5$, visualized, Epoch 1

Dropout

Example

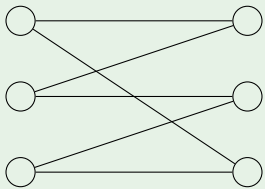

Figure: Dropout $\pi = 0.5$, visualized, Epoch 2

Dropout

Implementation

» Why?

- Dropout forces the network to learn redundancies

Dropout

Implementation

- Why?
  - Dropout forces the network to learn redundancies
- Implementation
  - In Keras, dropout is realized as additional layer
  - Applies to the layer before the dropout layer

```
model.add(layers.Dense(10)) # no edges dropped
model.add(layers.Dense(20)) # edges are dropped here
model.add(layers.Dropout(0.5))
```

Section 5

Exercise