# Planen and Evaluieren von Machine Learning Experimenten

Marina Sedinkina
Folien von Benjamin Roth

CIS LMU München

# Übersicht

1. Entwickeln von maschinellen Lernverfahren
   - Verbessern eines machine-learning Algorithmus
   - Aufteilung der Daten
   - Hyperparameter Optimization
   - Underfitting und Overfitting Erkennen
   - Datenmenge und Learning Curves

2. Zusammenfassung und praktische Tipps

# Übersicht

# Übersicht

# Verbessern eines machine-learning Algorithmus

- Welches Verfahren / Setup sollte man als erstes implementieren?
- Was könnte man versuchen, um davon ausgehend bessere Ergebnisse zu erzielen?

# Verbessern eines machine-learning Algorithmus

- Welches Verfahren / Setup sollte man als erstes implementieren?
  - So einfach wie möglich.
  - Wenige Annahmen.
  - Schnell zu implementieren.
- Was könnte man versuchen, um davon ausgehend bessere Ergebnisse zu erzielen?
  - Mehr Daten
  - Weniger Features (Mindestvorkommen)
  - Zusätzliche Features
  - Andere Repräsentation der Features (Featurekombinationen, ...)
  - Mehr/Weniger Regularisierung, andere Hyperparameter

# Verbessern eines machine-learning Algorithmus

- Klassifizierung von Email Spam:
  Angenommen, es wurde ein Klassifikator (SVM) auf den Trainingsdaten trainiert
- Auf einem neuen Datensatz ergibt sich eine hohe Fehlerquote.
- Was sollte als nächstes ausprobiert werden?
  - Mehr Daten
  - Weniger Features (Mindestwortfrequenz)
  - Zusätzliche Features (linguistische Analyse)
  - Andere Repräsentation der Features (Featurekombinationen, ...)
  - Mehr/Weniger Regularisierung, andere Hyperparameter

# Verbessern eines machine-learning Algorithmus

- Klassifizierung von Email Spam:
  Angenommen, es wurde ein Klassifikator (SVM) auf den Trainingsdaten trainiert
- Auf einem neuen Datensatz ergibt sich eine hohe Fehlerquote.
- Was sollte als nächstes ausprobiert werden?
  - Mehr Daten
  - Weniger Features (Mindestwortfrequenz)
  - Zusätzliche Features (linguistische Analyse)
  - Andere Repräsentation der Features (Featurekombinationen, ...)
  - Mehr/Weniger Regularisierung, andere Hyperparameter
- Manche dieser Optionen sind für sich genommen bereits ein umfangreiches Projekt.
- Die Frage wie weiter vorgegangen wird, sollte systematisch entschieden werden! (nicht intuitiv!)

# Fehlerdiagnostik

- Diagnostik: Verfahren um herauszufinden, was funktionieren könnte, und was nicht.
- Orientierungshilfe, wie die Vorhersagequalität eines machine learning-Algorithmus' verbessert werden könnte.
- Die Implementierung einer Fehlerdiagnostik nimmt Zeit in Anspruch.

# Performanz-Maße

- Ein Performanz-Maß ermöglicht die Vorhersagequalität eines Algorithmus quantitativ festzustellen
- Welches Maß verwendet werden kann, hängt von der Art der Aufgabe ab:
  - Klassifikation: Accuracy, F1-Score
  - Ranking: Mean Average Precision
  - Regression: Mean Squared Error
  - Spezialmaße: BLEU, ...

# Übersicht

# Aufteilung der Daten

- Erster Ansatz: Daten in Trainings und Testdatensatz aufteilen.

| Email Betreff | Label |
|---|---|
| y 2 k - texas log | 1 |
| emerging small cap | 0 |
| re : patchs work better then pillz | 0 |
| meter 1431 - nov 1999 | 1 |
| re : lyondell citgo | 1 |
| dobmeos with hgh my energy level has gone up | 0 |
| re : entex transistion | 1 |
| your prescription is ready . . oxwq s f e | 0 |
| get that new car 8434 | 0 |
| entex transistion | 1 |
| unify close schedule | 1 |
| await your response | 0 |

# Auswahl eines Modells

- Folgende Modelle werden durchprobiert:
  - 100 Merkmale
  - 1000 Merkmale
  - 10000 Merkmale
  - ...
- Option 1: Optimiere Parameter für jedes der Modelle (anhand Trainingsset), und wähle Model anhand des Performanz-Maßes auf dem Test-set.
- Angenommen das Modell mit 1000 Merkmalen gibt das beste Ergebnis.
- Ist das Performanz-Maß auf den Testdaten eine korrekte Schätzung der in Zukunft zu erwartenden Performanz?

# Auswahl eines Modells

- Folgende Modelle werden durchprobiert:
  - 100 Merkmale
  - 1000 Merkmale
  - 10000 Merkmale
  - ...
- Option 1: Optimiere Parameter für jedes der Modelle (anhand Trainingsset), und wähle Model anhand des Performanz-Maßes auf dem Test-set.
- Angenommen das Modell mit 1000 Merkmalen gibt das beste Ergebnis.
- Ist das Performanz-Maß auf den Testdaten eine korrekte Schätzung der in Zukunft zu erwartenden Performanz?
- Antwort: Nein. Der zusätzliche Parameter "Anzahl der Merkmale" ist auf das Testset überangepasst.

# Auswahl eines Modells

- Besser: Daten in Trainings-, Kreuzvalidierungs- and Testdaten aufteilen (z.B. 60%–20%–20%).
- Kreuzvalidierungsdaten werden auch Entwicklungsdaten genannt (cross-validation set, development set).

| Subject | Label |
|---|---|
| y 2 k - texas log | 1 |
| emerging small cap | 0 |
| re : patchs work better then pillz | 0 |
| meter 1431 - nov 1999 | 1 |
| re : lyondell citgo | 1 |
| dobmeos with hgh my energy level has gone up | 0 |
| re : entex transistion | 1 |
| your prescription is ready . . oxwq s f e | 0 |
| get that new car 8434 | 0 |
| entex transistion | 1 |
| unify close schedule | 1 |
| await your response | 0 |

# Trainings- / Kreuzvalidierungs- / Test-Fehler

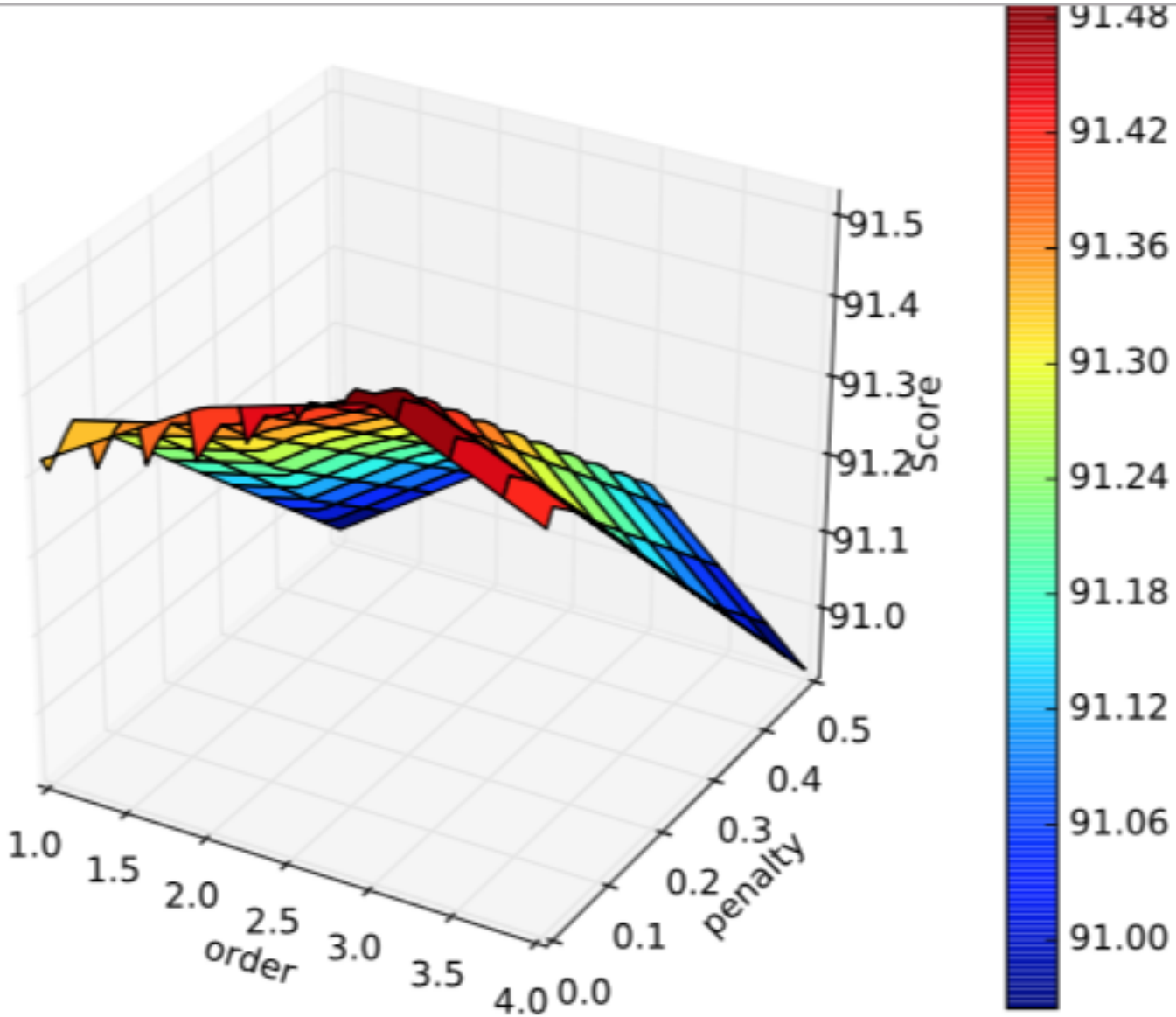
- Merkmalsgewichte werden auf Trainingsdaten geschätzt.
- Das Modell (Merkmale, Hyperparameter) wird anhand der Kreuzvalidierungsdaten ausgewählt.
- Die zu erwartende Performanz des Models wird anhand der Testdaten ermittelt.
- Ergebnisse auf Trainings- oder Kreuzvalidierungsdaten können nicht als Berwertung des Algorithmus aufgefasst werden!

# Übersicht

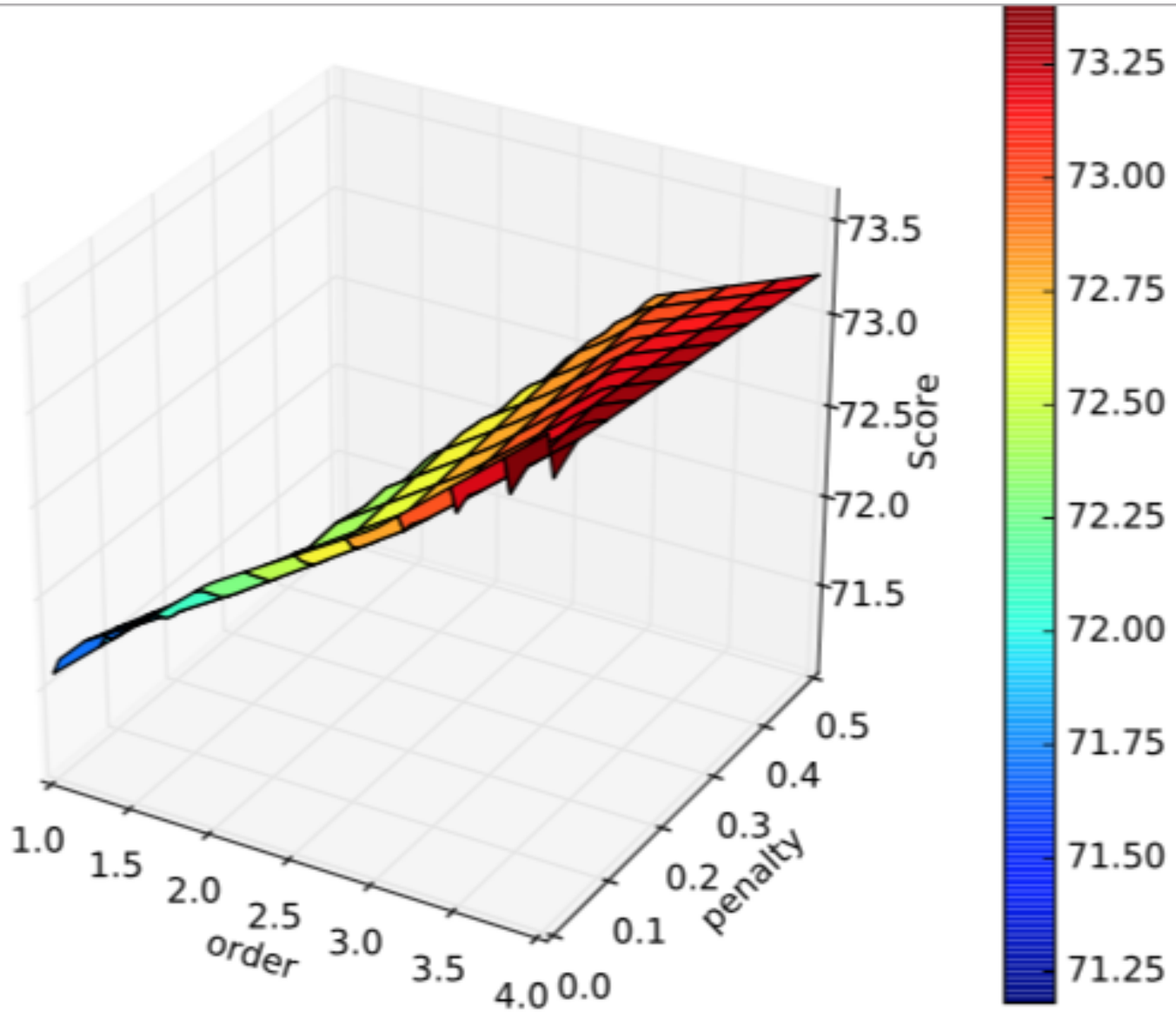
# Hyperparameter Optimization

- Richtige Hyperparameter auszuwählen ist ein Teil des Modelltrainings!
- Viele Hyperparameter:
  - embedding_size $\in \{10, ..., 1000\}$
  - hidden_size $\in \{10, ..., 1000\}$
  - l1_regularizer $\in \{10, ..., 1000\}$
  - l2_regularizer $\in \{10, ..., 1000\}$
  - dropout $\in \{10, ..., 1000\}$
  - optimizer $\in \{\textit{rmsprop}, \textit{adagrad}, \textit{sgd}\}$
  - ...
  - window size $\in \{1, 2, 3\}$
- Ansatz 1: Grid-Suche, geschachtelte for-Schleife: Probiere alle möglichen Wertekombinationen aus und wähle die beste Kombination anhand des Validierungsdatensatzes aus.
- Problem: der Suchraum über alle mögliche Kombinationen wird zu groß

# Beispiel für English: Hyperparameter Optimization



# Beispiel für Deutsch: Hyperparameter Optimization



# Hyperparameter Optimization 2

- Ansatz 2, Sampling: Für jeden der verschiedenen Parameter wähle Werte aus und führe so viele Konfigurationen aus, wie du dir leisten kannst (z. B. 100). Wähle Model anhand des Performanz-Maßes auf dem Entwiscklungset aus.

# Hyperparameter Optimization 2

- Ansatz 2, Sampling: Für jeden der verschiedenen Parameter wähle Werte aus und führe so viele Konfigurationen aus, wie du dir leisten kannst (z. B. 100). Wähle Model anhand des Performanz-Maßes auf dem Entwiscklungset aus.
- Intuition:

# Hyperparameter Optimization 2

- Ansatz 2, Sampling: Für jeden der verschiedenen Parameter wähle Werte aus und führe so viele Konfigurationen aus, wie du dir leisten kannst (z. B. 100). Wähle Model anhand des Performanz-Maßes auf dem Entwiscklungset aus.
- Intuition:
  - Einige Parameter sind unabhängig von den anderen Parameter. Es ist wichtiger, gute Werte für diese Parameter auszuwählen, als alle Kombinationen auszuprobieren.

# Hyperparameter Optimization 2

- Ansatz 2, Sampling: Für jeden der verschiedenen Parameter wähle Werte aus und führe so viele Konfigurationen aus, wie du dir leisten kannst (z. B. 100). Wähle Model anhand des Performanz-Maßes auf dem Entwiscklungset aus.
- Intuition:
  - Einige Parameter sind unabhängig von den anderen Parameter. Es ist wichtiger, gute Werte für diese Parameter auszuwählen, als alle Kombinationen auszuprobieren.
  - Einige Parameter verbessern die Ergebnisse nur ein bisschen. Verschwende keine Zeit, diese Parameter zu untersuchen. Untersuche andere Parameter.

# Hyperparameter Optimization 2

- Ansatz 2, Sampling: Für jeden der verschiedenen Parameter wähle Werte aus und führe so viele Konfigurationen aus, wie du dir leisten kannst (z. B. 100). Wähle Model anhand des Performanz-Maßes auf dem Entwiscklungset aus.
- Intuition:
  - Einige Parameter sind unabhängig von den anderen Parameter. Es ist wichtiger, gute Werte für diese Parameter auszuwählen, als alle Kombinationen auszuprobieren.
  - Einige Parameter verbessern die Ergebnisse nur ein bisschen. Verschwende keine Zeit, diese Parameter zu untersuchen. Untersuche andere Parameter.
- Praktische Tipps:

# Hyperparameter Optimization 2

- Ansatz 2, Sampling: Für jeden der verschiedenen Parameter wähle Werte aus und führe so viele Konfigurationen aus, wie du dir leisten kannst (z. B. 100). Wähle Model anhand des Performanz-Maßes auf dem Entwiscklungset aus.
- Intuition:
  - Einige Parameter sind unabhängig von den anderen Parameter. Es ist wichtiger, gute Werte für diese Parameter auszuwählen, als alle Kombinationen auszuprobieren.
  - Einige Parameter verbessern die Ergebnisse nur ein bisschen. Verschwende keine Zeit, diese Parameter zu untersuchen. Untersuche andere Parameter.
- Praktische Tipps:
  - E.g.: 0.01, 0.1, 1, 10, 100
    or: 0.01, 0.02, 0.1, 0.3, 1,3, 10, 30, 100
    or: 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100

# Hyperparameter Optimization 3

- Ansatz 2a: Mache Sampling nur über einen kleinen "sinnvollen" Bereich von Parameterwerten.

# Hyperparameter Optimization 3

- Ansatz 2a: Mache Sampling nur über einen kleinen "sinnvollen" Bereich von Parameterwerten.
  - Falls einige Werte an der Grenze liegen:

# Hyperparameter Optimization 3

- Ansatz 2a: Mache Sampling nur über einen kleinen "sinnvollen" Bereich von Parameterwerten.
  - Falls einige Werte an der Grenze liegen:
  - Erweitere den Bereich an diesen Grenzen manuell (Konfigurationsdatei / den Code bearbeiten) und optimiere die Parameter erneut

# Hyperparameter Optimization 3

- Ansatz 2a: Mache Sampling nur über einen kleinen "sinnvollen" Bereich von Parameterwerten.
  - Falls einige Werte an der Grenze liegen:
  - Erweitere den Bereich an diesen Grenzen manuell (Konfigurationsdatei / den Code bearbeiten) und optimiere die Parameter erneut
- Ansatz 2b: Ansatz 2a automatisieren

# Hyperparameter Optimization 3

- Ansatz 2a: Mache Sampling nur über einen kleinen "sinnvollen" Bereich von Parameterwerten.
  - Falls einige Werte an der Grenze liegen:
  - Erweitere den Bereich an diesen Grenzen manuell (Konfigurationsdatei / den Code bearbeiten) und optimiere die Parameter erneut
- Ansatz 2b: Ansatz 2a automatisieren
  - Erweitere den Teilbereich, wenn der beste Wert nach n Iterationen auf einer Grenze liegt

# Hyperparameter Optimization 3

- Ansatz 2a: Mache Sampling nur über einen kleinen "sinnvollen" Bereich von Parameterwerten.
  - Falls einige Werte an der Grenze liegen:
  - Erweitere den Bereich an diesen Grenzen manuell (Konfigurationsdatei / den Code bearbeiten) und optimiere die Parameter erneut
- Ansatz 2b: Ansatz 2a automatisieren
  - Erweitere den Teilbereich, wenn der beste Wert nach n Iterationen auf einer Grenze liegt
- Mehr Ideen: *Yoshua Bengio, "Practical recommendations for gradient-based training of deep architectures"*

# Übersicht

# Feststellen von Bias oder Variance

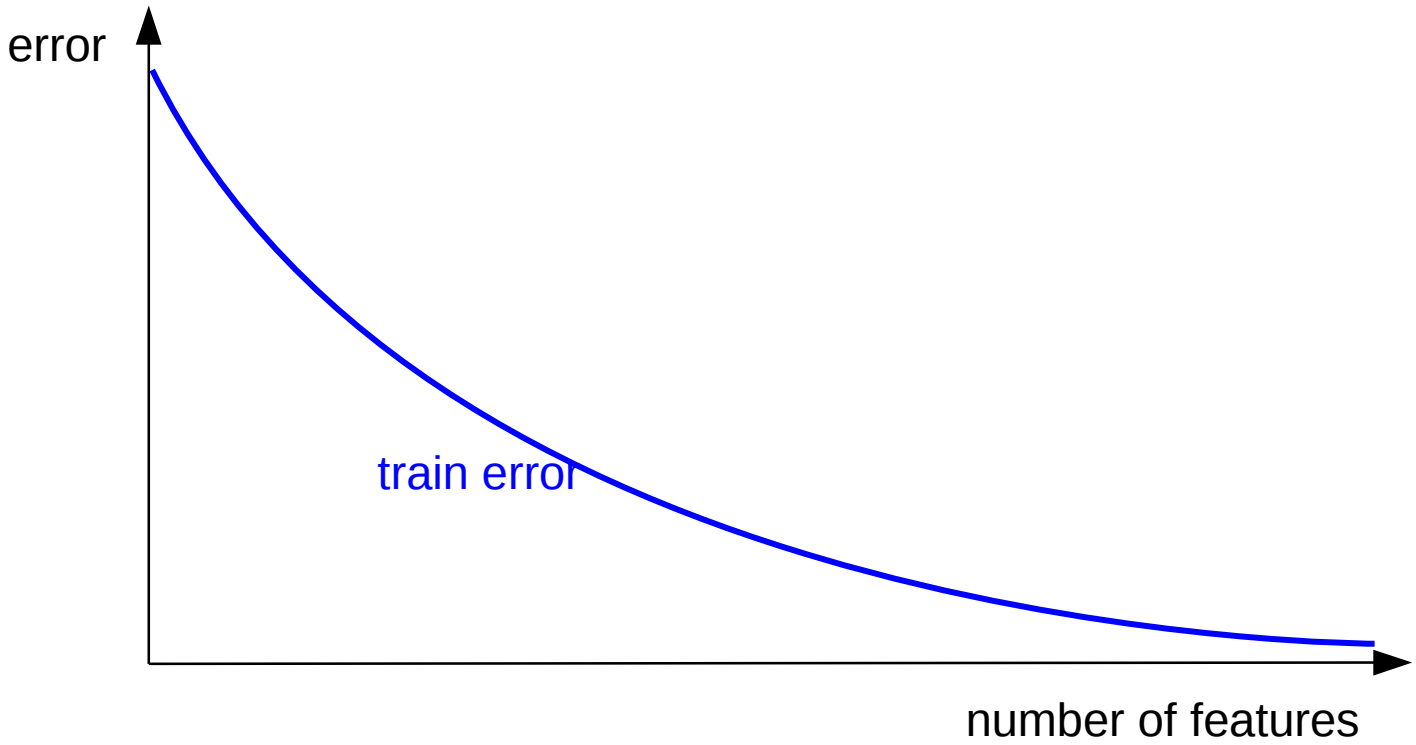
- Bias: Unteranpassung an Trainingsdaten (underfit).
  - Modell nicht “mächtig” genug.

- Variance: Überanpassung an Trainingsdaten (overfit).

# Feststellen von Bias oder Variance

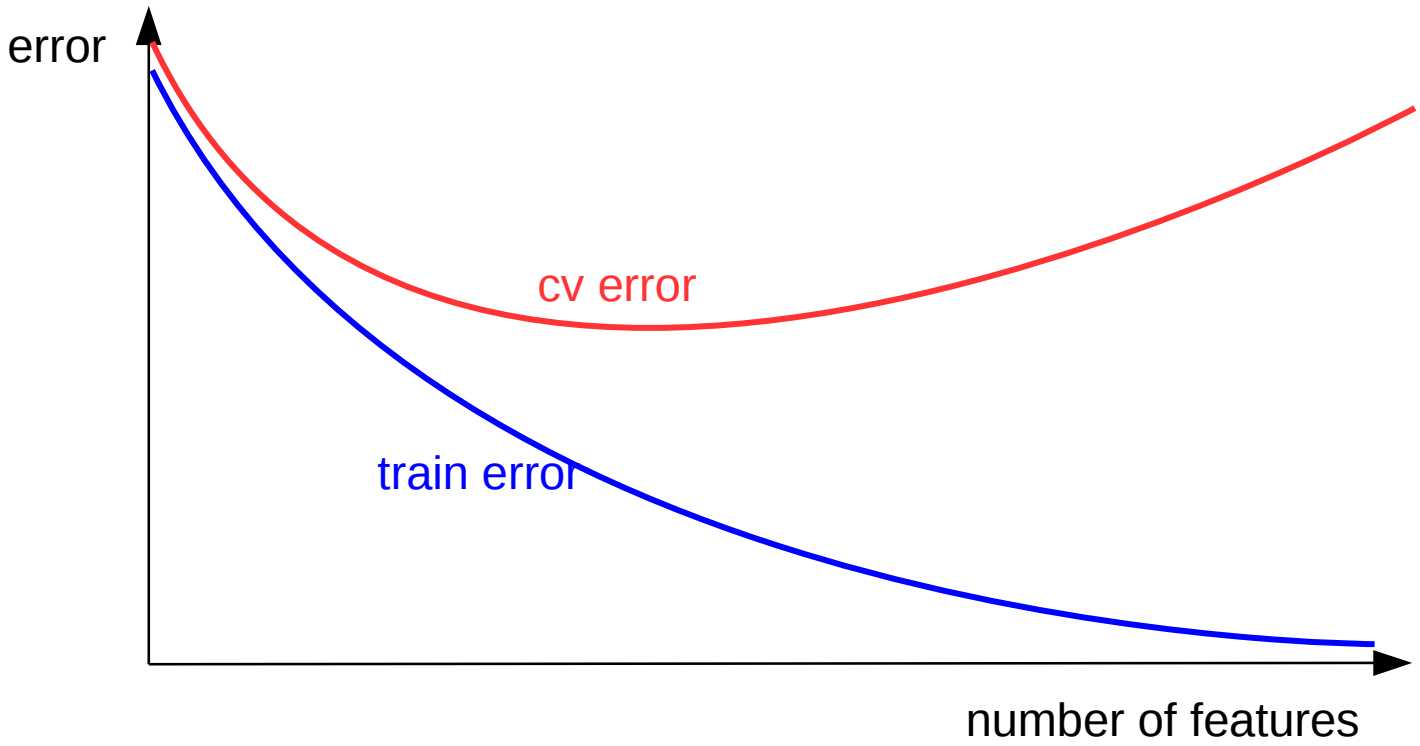
- Bias: Unteranpassung an Trainingsdaten (underfit).
  - Modell nicht "mächtig" genug.
  - Zu wenige Merkmale?
  - Zu viel Regularisierung?
- Variance: Überanpassung an Trainingsdaten (overfit).
  - Zu viele Parameter/Merkmale?
  - Zu wenig Regularisierung?
  - Zu wenige Daten?

# Fehlerrate bei Erhöhen der Modellkapazität



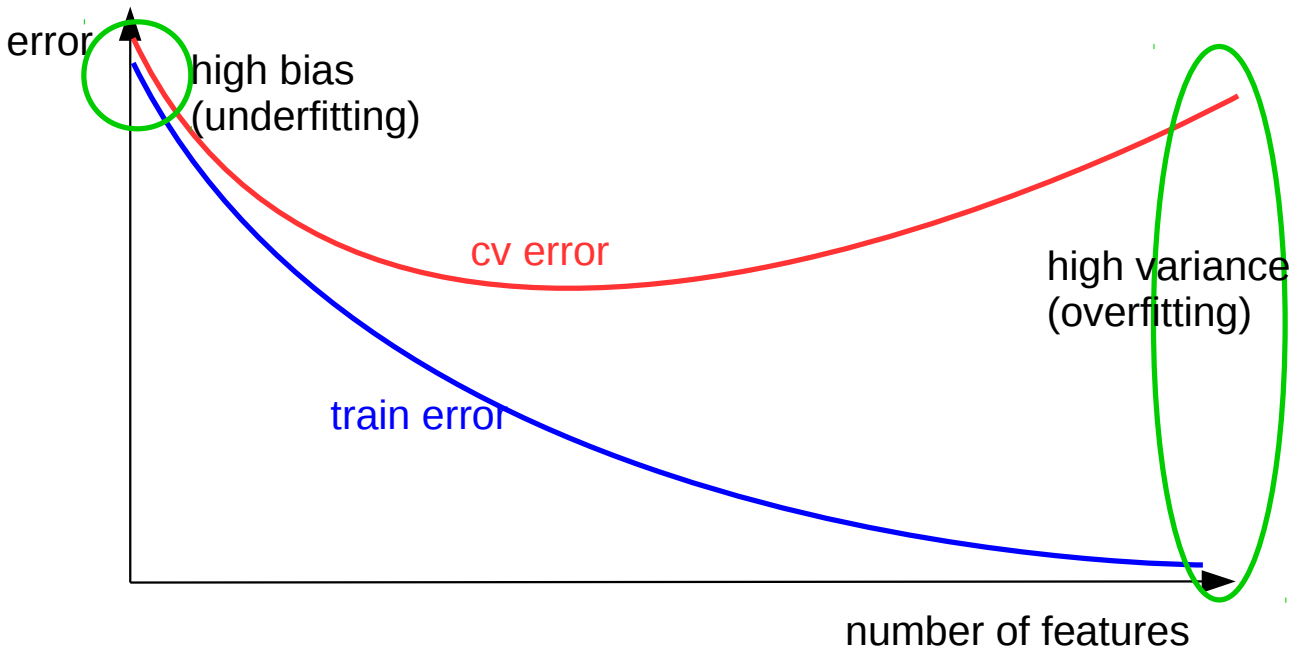
# Diagnose: Underfitting oder Overfitting?

- Angenommen der Crossvalidierungsfehler is groß.
- Ist es ein Bias- (underfitting) oder Variance- (overfitting) Problem?



# Diagnose: Underfitting oder Overfitting?

- Angenommen der Crossvalidierungsfehler is groß.
- Ist es ein Bias (underfitting) oder Variance (overfitting) Problem?



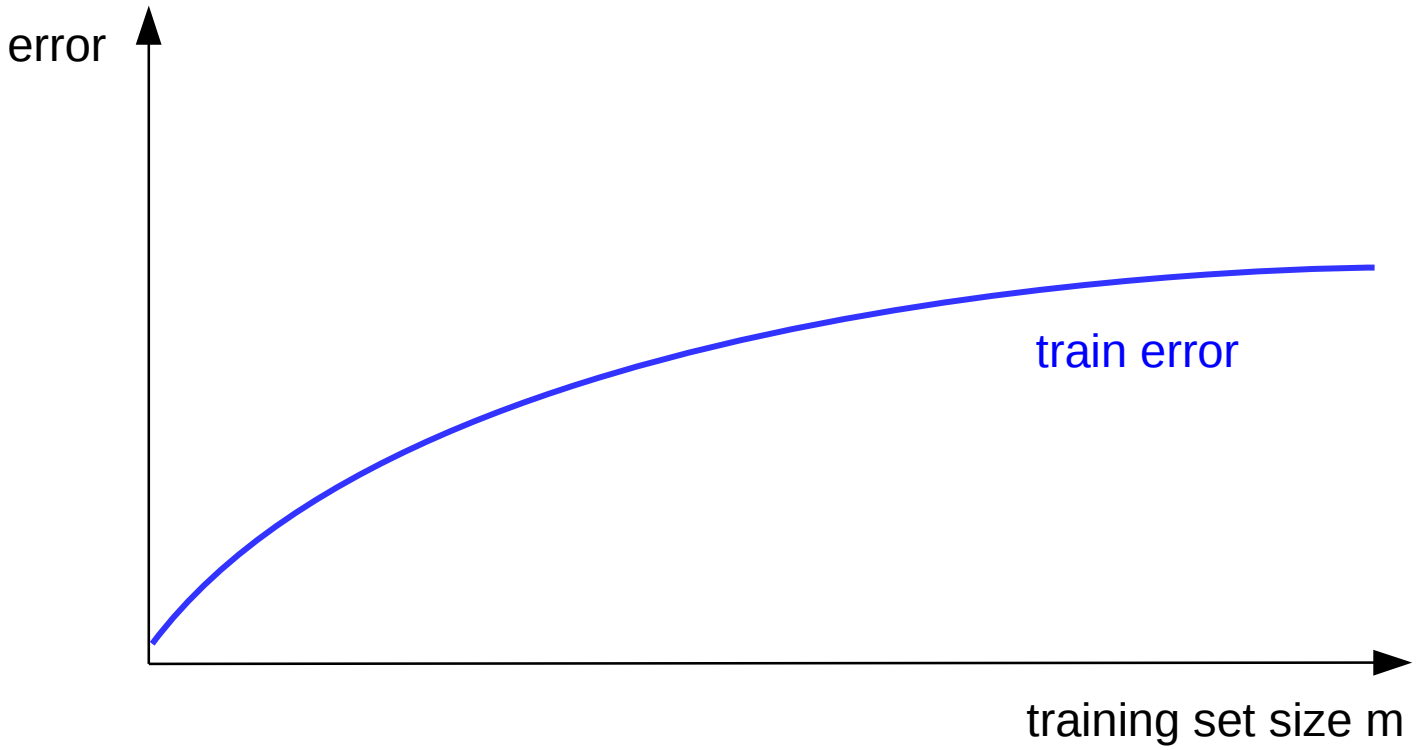
# Diagnose: Underfitting oder Overfitting

- Bias (underfitting):
  - train error hoch
  - cv error $\approx$ train error
- Variance (overfitting):
  - train error niedrig
  - cv error $>>$ train error

# Übersicht

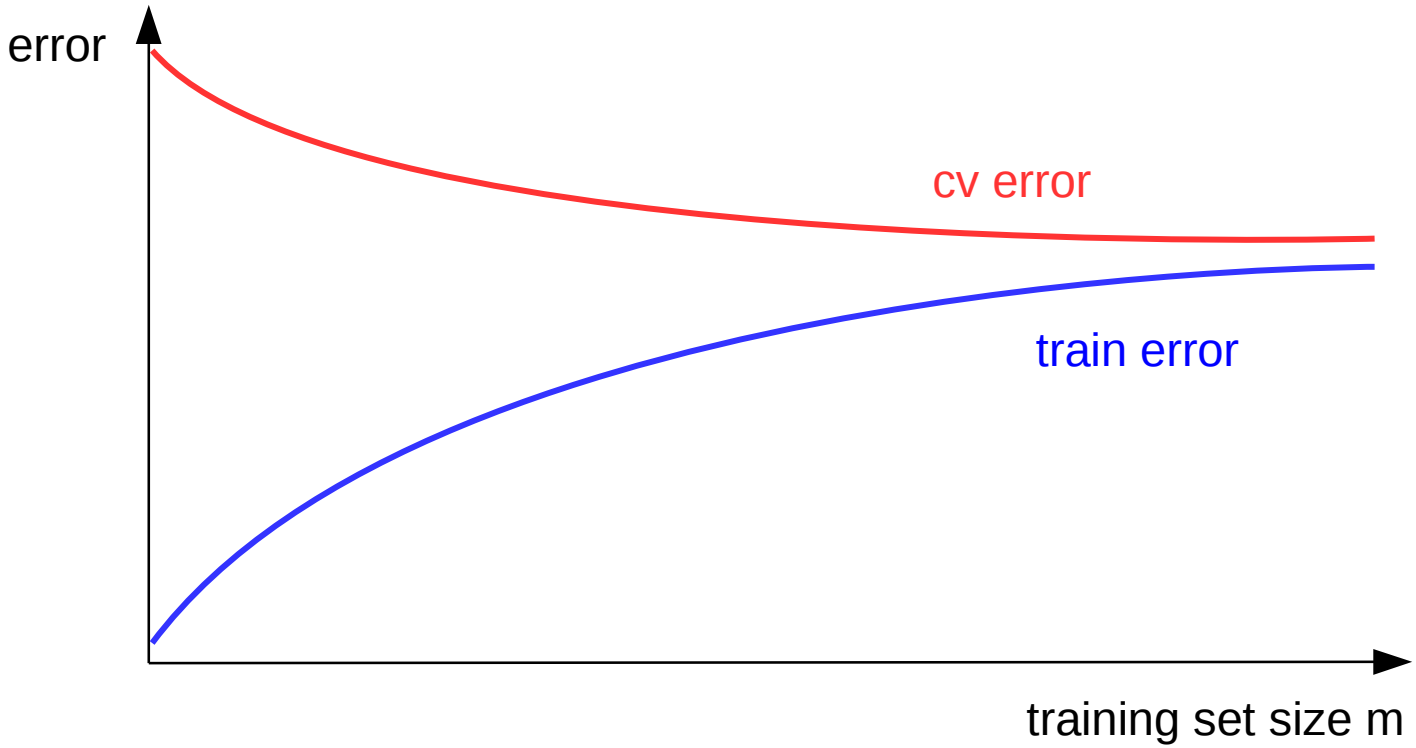
# Learning Curves

- "Learning Curve": Fehlerfunktion in Abhängigkeit von der Datenmenge.
- Je mehr Daten im Training vorhanden sind, desto schwieriger ist es ein Modell zu finden, dass alle Trainingsdaten perfekt modelliert ...



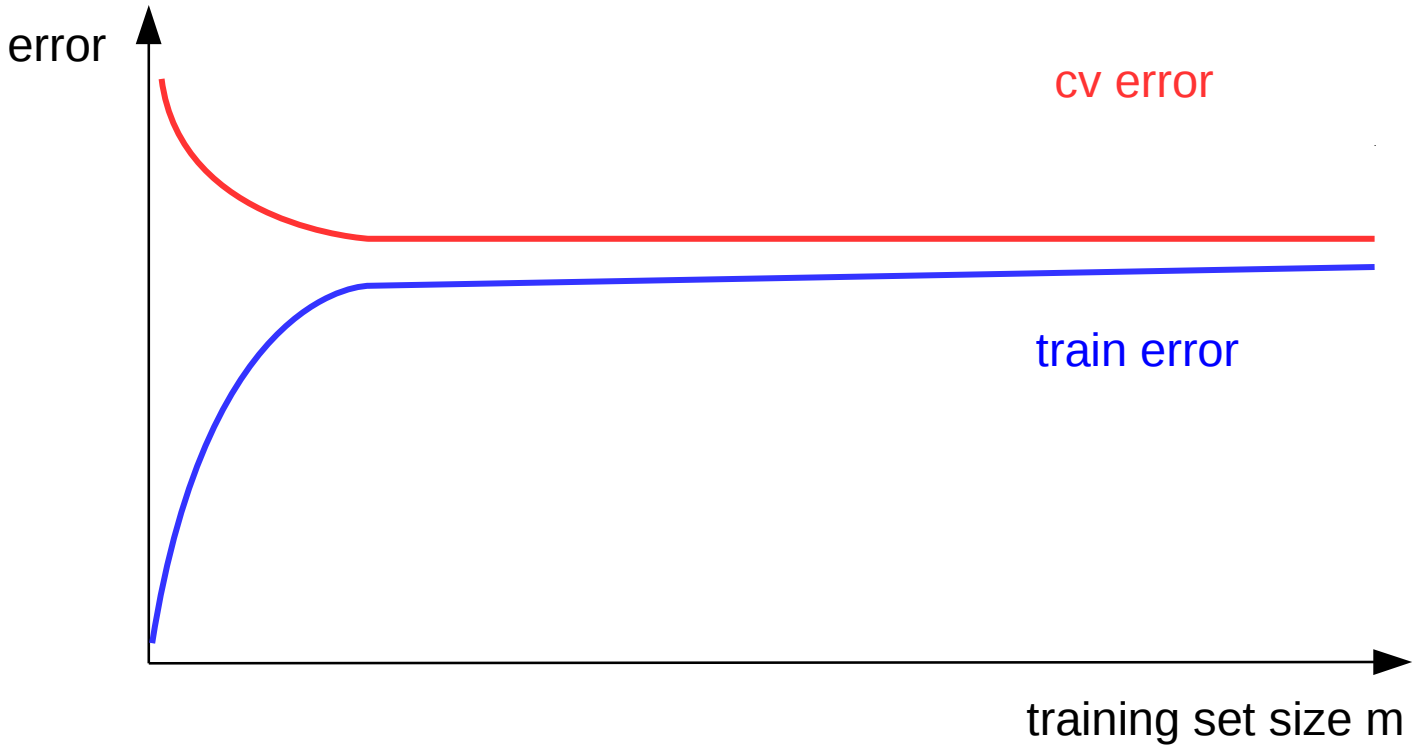
# Learning Curves

- "Learning Curve": Fehlerfunktion in Abhängigkeit von der Datenmenge
- Je mehr Daten im Training vorhanden sind, desto schwieriger ist es ein Modell zu finden, dass alle Trainingsdaten perfekt modelliert ...
- ... jedoch steigt bei mehr Trainingsdaten die Qualität der Vorhersage für ungesehene Daten.



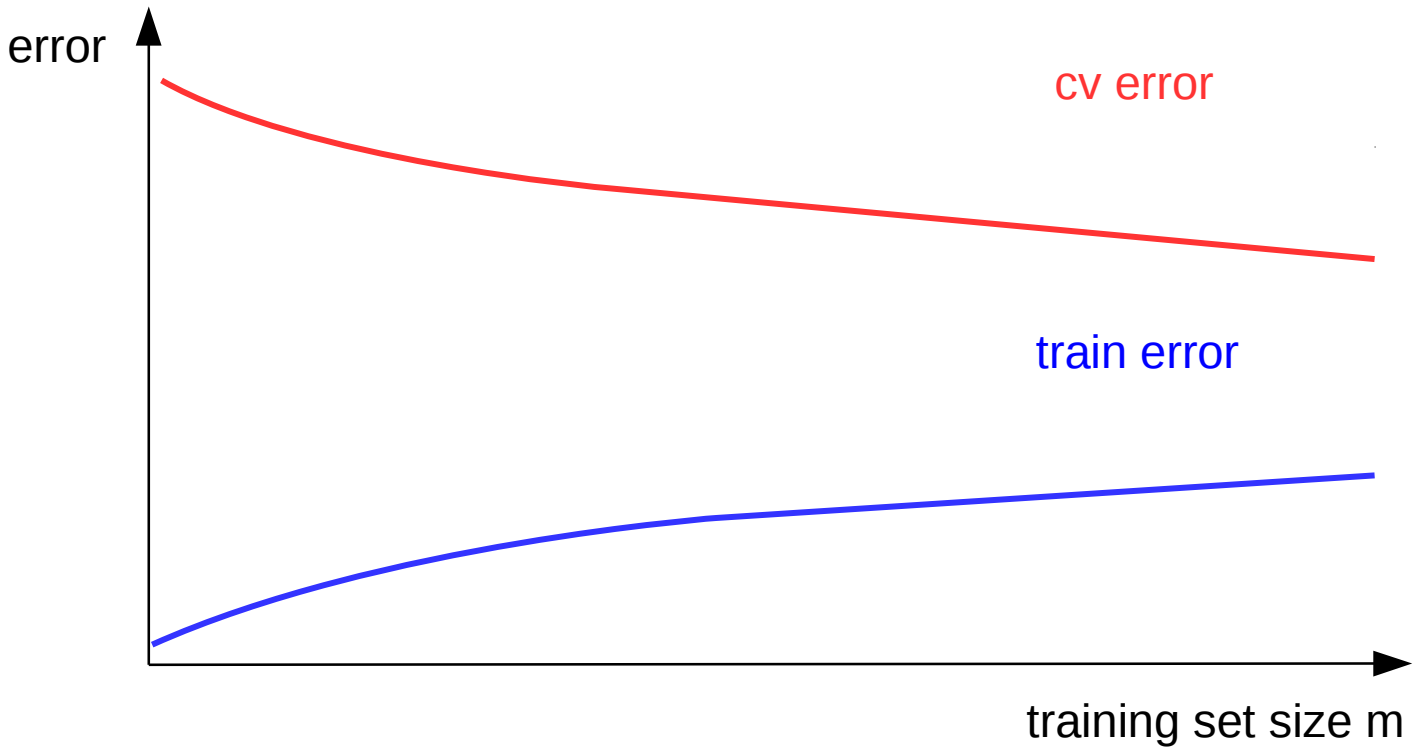
# Learning Curves bei underfitting-Modellen

- Bei underfitting-Modellen ändert sich der Fehler in Abhängigkeit von zusätzlichen Daten nicht wesentlich.



# Learning Curves bei overfitting-Modellen

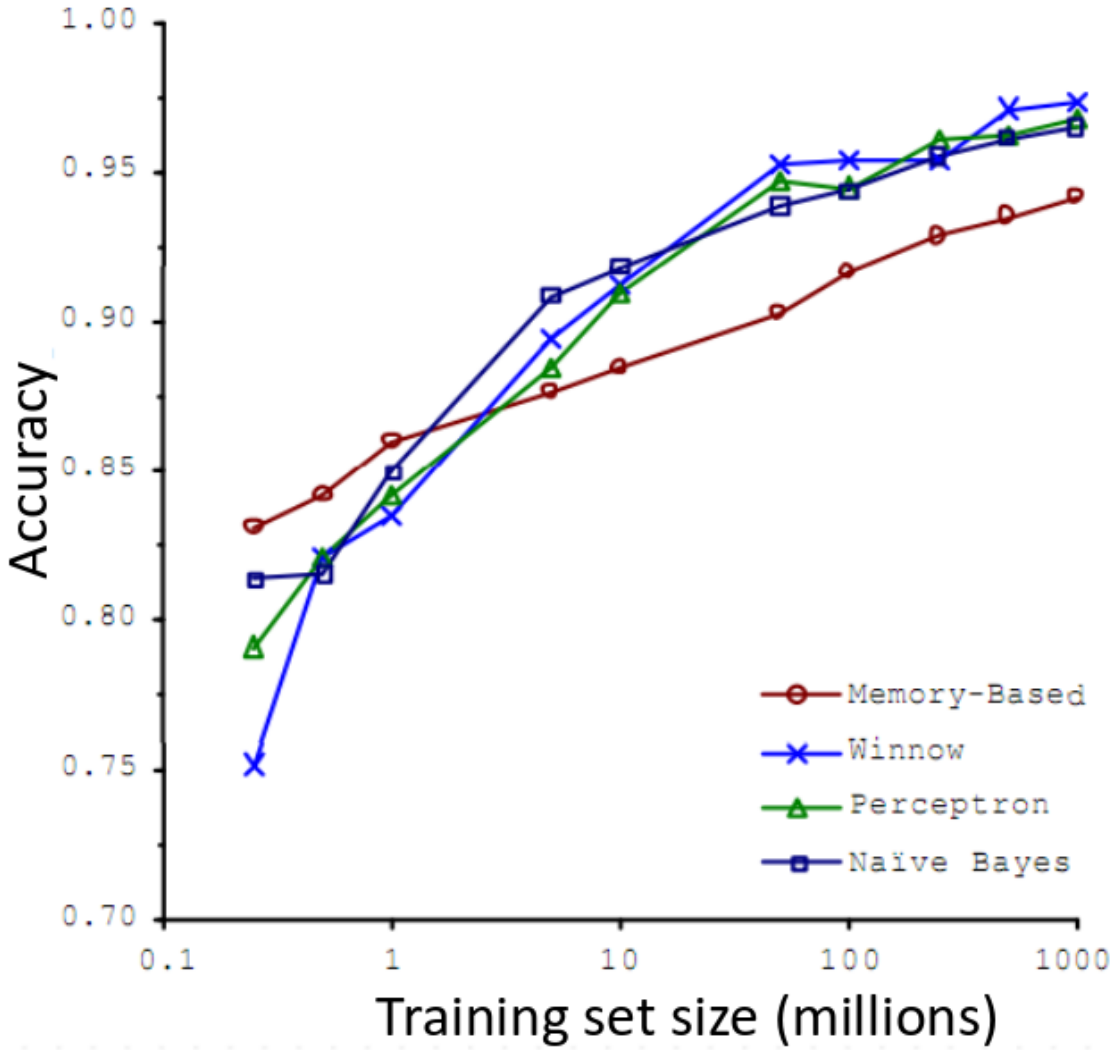
- Großer Unterschied zwischen Trainings- und Testfehler.
- Zusätzliche Trainingsdaten reduzieren den Testfehler.
- Zusätzliche Trainingsdaten erhöhen den Trainingsfehler (weniger Overfitting)



# Übersicht

# Sind mehr Daten immer besser?

- Daten zu gewinnen ist mit Aufwand verbunden.
- Wann lohnt sich dieser Aufwand?




[Banko & Brill, 2001]

# Sind mehr Daten immer besser?

- Banko and Brill 2001: "It's not who has the best algorithm that wins. It's who has the most data."
- Annahmen:
  - Merkmale enkodieren alle wesentlichen Informationen, so dass ein Mensch die Entscheidung souverän treffen könnte.
  - Der Lernalgorithmus hat eine hohe Kapazität (hohe Varianz, overfitting).
- Unter diesen Annahmen ist es eine gute Idee, mehr Daten zu gewinnen.
- Ansonsten ist es vielversprechender, an Merkmalen und Algorithmus zu arbeiten.

# Zusammenfasssung: Verbessern von Performanz

- Ausgangssituation: Klassifikator hat zu große Fehlerrate auf Kreuzvalidierungsdaten.
- Diagnostik: Learning Curves
  - Testfehler und CV-Fehler für 10%, 20%, ... 100% der Testdaten anzeigen.
  - $\Rightarrow$ Overfitting oder Underfitting.
- Nächste Schritte:
  - Problem ist Overfitting:
    - ★ Regularisierung erhöhen
    - ★ Weniger Merkmale
    - ★ Mehr Trainingsdaten
  - Problem ist Underfitting:
    - ★ Regularisierung erniedrigen
    - ★ Merkmalskombinationen
    - ★ Zusätzliche Merkmale

# Fehleranalyse

- Beginne mit einem einfachen Algorithmus, der schnell implementiert werden kann.
- Auf Kreuzvalidierungsdaten testen und Hyperparameter optimieren.
- Learning Curves anzeigen, um zu sehen ob mehr Daten oder mehr Features helfen könnten.
- Fehleranalyse:
  - Von Hand Beispiele in den Kreuzvalidierungsdaten suchen, bei denen der Algorithmus Fehler gemacht hat.
  - Gibt es systematische Fehler?
- Falls das Problem Underfitting war, neue Features anhand der Beobachtungen konstruieren.
- Falls das Problem Overfitting war, Features anhand der Beobachtungen generalisieren (oder neue Daten gewinnen).

# Fehleranalyse: Spam-Email Beispiel

- 500 Beispiele in Kreuzvalidierungsset
- 100 falsch klassifiziert
- Durchsehen und von Hand kategorisieren:
- Welche Art von Email:
  - Pharma
  - Gefälschte Produkte
  - Fishing-emails
  - andere
- Welche Features könnten helfen:
  - Länge der Email
  - Beabsichtigte 5chreibfehl3r
  - andere

# Noch Fragen?