Physics 361 - Machine Learning in Physics

Lecture 5 – Basics of Machine Learning

Feb. 6th 2024

Moritz Münchmeyer
Unit 2: Machine Learning Basics

2.1 Machine Learning concepts using the example of Linear Regression (cont.)
Linear regression (cont'd)

Very simple ML model:
\[
\hat{y} = \hat{w}^T \hat{x}
\]

\( \hat{x} \): input
\( \hat{y} \): output

Training data:
\[
\hat{x}_{\text{train}} = \begin{pmatrix} \hat{x}_1 \hat{x}_2 \ldots \hat{x}_n \end{pmatrix} \text{\{examples\}}
\]
\( \hat{y}_{\text{train}} \): desired output = label
\( \hat{y} \): estimate

Loss function:
\[
\text{MSE} = \frac{1}{m} \sum_{i=1}^{m} \| \hat{y}_i - \hat{y}_i \|^2
\]

Can be minimized w.r.t. to \( \hat{w} \):
\[
\hat{w} = \left( \hat{x}_{\text{train}}^T \hat{x}_{\text{train}} \right)^{-1} \hat{x}_{\text{train}}^T \hat{y}_{\text{train}}
\]
Last slide: \( y \) was a 1-dimensional output. We can also have multidimensional regression targets:

\[
\vec{y} = M \vec{x} + b
\]

(vector output) (before we had \( y = \hat{w}^T \vec{x} \))

- These linear transformations are called Linear Layers in machine learning.
- Side note: While linear models are analytically solvable (for invertible \( M \)), for very high dimensional problems it is more efficient to solve them by optimization.
**Polynomial regression**

In polynomial regression we fit a higher order polynomial.

E.g. 2nd order pol. \( \hat{y} = b + w_1 x + w_2 x^2 \)

General: \( \hat{y} = b + \sum_{i=1}^{n} w_i x_i \)

This model can be solved for \( \hat{w} \) with the same equations as linear regression because it is still linear in the parameters \( w_i \).
Capacity, Overfitting, Underfitting

* How well does a model trained/fitted to the training data work on novel data.
  
  \[\Rightarrow\text{Generalization Error}\]

* We usually assume that training data and test data are drawn from the same distribution.
  \[\text{i.i.d.} \text{ data: identically distributed}\]

* Generalization is closely related to overfitting and underfitting.

  \[\text{underfit: training error is large but test error is small}\]
  \[\text{overfit: training error is small but test error is large}\]
Example of polynomial regression:

- The space of functions that the model can represent is called the capacity.
- Occam's razor: choose the simplest that fits.
- Lower capacity $\rightarrow$ tends to generalize better.
- Higher capacity $\rightarrow$ reduces training error.

$\Rightarrow$ Need to optimize the capacity or regularize.
Typical behavior of error vs capacity (after training out the model)

We could e.g. train models with various numbers of neural network layers. Usually this is not the main approach. Instead we use "regularization" to avoid overfitting.
Regularization

Idea: keep the model capacity fixed, but encourage simpler solutions.

Common method: weight decay

New loss to minimize: $J(w) = \text{MSE}_{\text{train}} + \lambda \sum w^2$

$L-2$ norm

$L-2$ regularization

This discourages large weights
L-2 regularization is one of several popular methods of regularization.

Regularization: any method meant to prevent overfitting without changing the model capacity.

Two other popular methods which we discuss later:
- dropout
- early stopping.
Hyperparameters

- Hyperparameters are parameters that are NOT part of the fitting/optimization procedure.

- Examples:
  - Model architecture
    - e.g., maximum order of polynomials
    - depth of a neural network
  - Learning parameters
    - e.g., learning rate
  - Regularization parameters e.g. $\lambda$

- To choose the best hyperparameters one often splits off a validation data set from the training data (~20% thereof).
Relation of MSE and maximum likelihood

- In linear regression we were learning $y$ from $x$. Instead now we want to learn $P(y | x)$. \( \Rightarrow \) we get an error bar.

- If we assume that the error is Gaussian we want to learn: \( P(y | x) = \mathcal{N}(y ; \hat{y}(x; \theta), \sigma^2) \)

- The loss is now the likelihood of the training data:

  \[
  \mathcal{L}(\theta) = \sum_{i=1}^{n} \log p(y_i | x_i; \theta) = -m \log \sigma - \frac{m}{2} \log (2\pi) - \sum_{i=1}^{n} \frac{1}{2} \left( \frac{(y_i - \hat{y}(x_i; \theta))^2}{\sigma^2} \right)
  \]

  - model parameters, here $\hat{\theta} = \hat{\omega}$ is linear regression.

- We want to maximize this with respect to $\theta$. Maximize $L$ is the same as minimizing MSE.
Course logistics

• Reading for this lecture:
  • For example: Deeplearningbook.org 5.

• Problem set: First problem set end of this week.