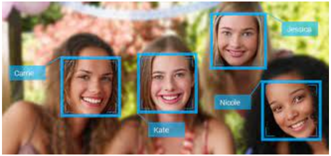
Machine learning for particle physicists

I. Linear models for regression and classification

Anja Butter

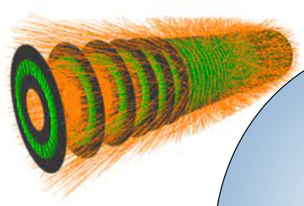
26th Vietnam School of Physics

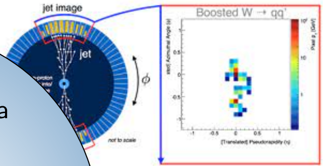
Machine Learning seems to be everywhere



The triumph of AlexNet in 2012 triggered a ML wave!

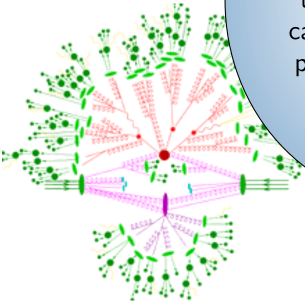
Why Machine Learning in particle physics?

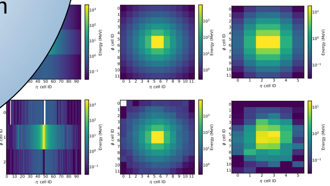




LHC = BIG data

applications:
jet calibration
track reconstruction
calorimeter simulation
particle identification
event generation
...





What is Machine Learning?

Formal definition

A computer program is said to learn from experience E
with respect to some class of tasks T and performance measure P
if its performance at tasks in T,
as measured by P,
improves with experience E.

Tom Michael Mitchell (1997)

What is Machine Learning?

Formal definition

A computer program is said to learn from experience

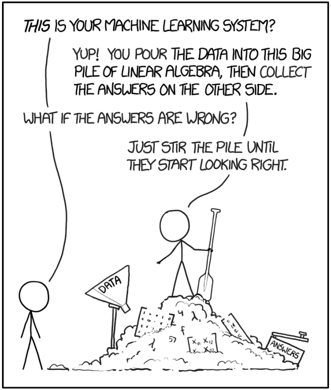
if its performance at tasks

improves with experience .

Tom Michael Mitchell (1997)

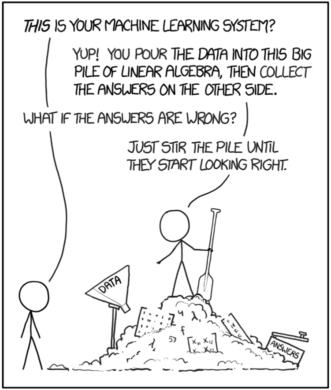
What is Machine Learning?

What it feels like (sometimes)



What is Machine Learning?

What it feels like (sometimes)



Aim of this lecture:
Giving you the tools for a systematic approach to use ML in particle physics

Can Machine Learning do anything?

A word of caution before we dive in

Thanks to machine-learning algorithms,
the robot apocalypse was short-lived.

The core of machine learning is to find structure in data - no more no less.
Beware the data prior!

How to become a ML expert?

http://nirvacana.com/thoughts/2013/07/08/becoming-a-data-scientist/

Luckily we are physicists :)

This lecture:
We will start with a simple problem and learn new things on a need-to-know basis

The following lectures build on each other.
$\rightarrow$If anything is unclear, please do not hesitate to ask!

Can we predict the signal of a calorimeter?

Setting up a machine learning problem

$$\text{Dataset } (x_i, y_i)_{i=1,\dots,n}$$

$\nearrow$ $\nwarrow$

feature — label

(particle E, id, ...) (signal)

$$\text{True function } f : X \to Y, f(x_i) = y_i$$

$$\text{Hypothesis } h \in H : X \to Y, f(x_i) = \tilde{y}_i \leftarrow \text{prediction}$$

Loss:

$$\mathcal{L}_{MSE} = \frac{1}{n} \sum_{i}^{n} (y_i - \tilde{y}_i)^2$$

(measure for goodness of approximation)

Learning: Minimization of the loss function

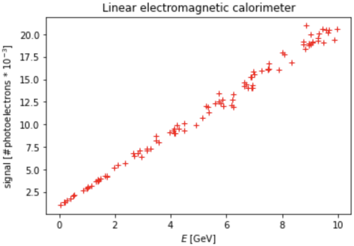
Can we predict the signal of a calorimeter?

Univariate linear regression

Predict signal y as function of energy x

dependent variable

regressor



Linear regression model:

$$\tilde{y}_i = w_1 x_i + w_2$$

with model parameters w_j

Find optimal linear model
$\rightarrow$ depends on data and chosen loss function

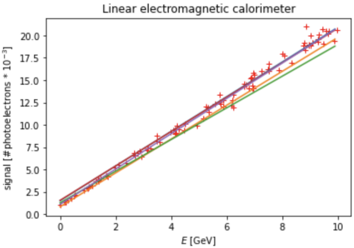
Can we predict the signal of a calorimeter?

Univariate linear regression

Predict signal y as function of energy x

dependent variable

regressor



Linear regression model:

$$\tilde{y}_i = w_1 x_i + w_2$$

with model parameters w_j

Find optimal linear model
$\rightarrow$ depends on data and chosen loss function

Optimizing a linear problem

Chosen loss: mean squared error

$$
\begin{aligned}
\mathcal{L}_{MSE} &= \frac{1}{n}\sum_{i}^{n}(y_i - \tilde{y}_i)^2 \\
&= \frac{1}{n}\sum_{i}^{n}\left(y_i - \begin{pmatrix} x_i \\ 1 \end{pmatrix}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}\right)^2 \quad \text{with } \begin{pmatrix} x_i \\ 1 \end{pmatrix} = \boldsymbol{x}_i, \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \boldsymbol{w}
\end{aligned}
$$

Optimizing a linear problem

Chosen loss: mean squared error

$$\begin{aligned}\mathcal{L}_{MSE} &= \frac{1}{n}\sum_{i}^{n}(y_i - \tilde{y}_i)^2 \\ &= \frac{1}{n}\sum_{i}^{n}\left(y_i - \begin{pmatrix} x_i \\ 1 \end{pmatrix}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}\right)^2 \quad \text{with } \begin{pmatrix} x_i \\ 1 \end{pmatrix} = \boldsymbol{x}_i, \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \boldsymbol{w}\end{aligned}$$

Minimizing the loss

$$\min_{w_j} \mathcal{L}_{MSE} \rightarrow \nabla_w \mathcal{L}_{MSE} = \sum_{i=1}^{n} 2\left(y_i - \boldsymbol{x}_i \boldsymbol{w}\right)\boldsymbol{x}_i \overset{!}{=} 0$$

Optimizing a linear problem

Chosen loss: mean squared error

$$\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i}^{n}(y_i - \tilde{y}_i)^2$$

$$= \frac{1}{n}\sum_{i}^{n}\left(y_i - \begin{pmatrix} x_i \\ 1 \end{pmatrix}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}\right)^2 \quad \text{with } \begin{pmatrix} x_i \\ 1 \end{pmatrix} = \boldsymbol{x}_i, \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \boldsymbol{w}$$

Minimizing the loss

$$\min_{w_j} \mathcal{L}_{MSE} \rightarrow \nabla_w \mathcal{L}_{MSE} = \sum_{i=1}^{n} 2\left(y_i - \boldsymbol{x}_i \boldsymbol{w}\right)\boldsymbol{x}_i \stackrel{!}{=} 0$$

$$\rightarrow \text{(I.)} \sum_{i=1}^{n}\left(y_i - \boldsymbol{x}_i \boldsymbol{w}\right) = 0 \qquad \text{(II.)} \sum_{i=1}^{n}\left(y_i - \boldsymbol{x}_i \boldsymbol{w}\right) x_i = 0$$

$$\boldsymbol{w}\sum_{i=1}^{n}\boldsymbol{x}_i = \sum_{i=1}^{n} y_i$$

Exercise

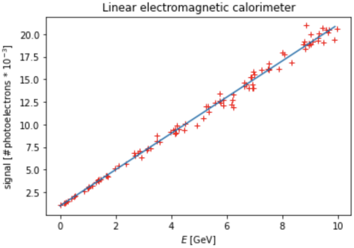
$$w_2 = \bar{y} - w_1 \bar{x}$$

$$w_1 = ?$$

Optimizing a linear problem

Exact analytic solution:

$$w_1 = \frac{\sum_{i=1}^{n} (y_i - \bar{y}) (x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$
$$w_2 = \bar{y} - w_1 \bar{x}$$



Python/numpy warm up exercise:
→ Implement this problem (git)

Multivariate linear regression

We can generalize our result to multidimensional feature vectors!
Adopting a more compact notation we write:

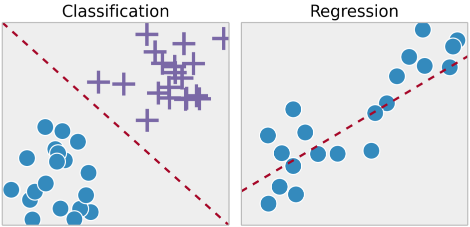
$$\tilde{\boldsymbol{y}} = \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_n \end{pmatrix} = \begin{pmatrix} x_{11}w_1 + x_{12}w_2 + \cdots + x_{1d}w_d \\ x_{21}w_1 + x_{22}w_2 + \cdots + x_{2d}w_d \\ \vdots \\ x_{n1}w_1 + x_{n2}w_2 + \cdots + x_{nd}w_d \end{pmatrix} = \boldsymbol{X}\boldsymbol{w}$$

Minimizing the loss yields

$$\nabla_{\boldsymbol{w}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||^2 = \sum_{i=1}^{n} 2\,(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})\,\boldsymbol{X} \stackrel{!}{=} 0$$

$$\rightarrow \boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Classification vs Regression



cats vs dogs
particle identification
top vs QCD jets
...?

smartphone price
jet energy scale
amplitude
...?

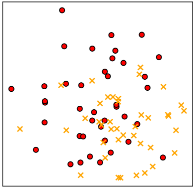
Can we predict the type of a jet

Setting up a classification problem

Dataset $(\boldsymbol{x}_i, y_i)_{i=1,\dots,n}$

feature $\nearrow$ $\nwarrow$ label

$(n_{particles}, m_j, \dots)$ $(0/1 \rightarrow \text{QCD/top})$



Hypothesis:
$h(\boldsymbol{x}_i) = \tilde{y}_i \in [0,1] \leftarrow$ prediction

Interpretation:
$h(\boldsymbol{x}_i) = P(y_i = 1 | \boldsymbol{x}_i) \leftarrow$ probability

Minimize probability that h assigns wrong class.

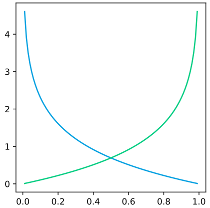
How to build a loss function for classification

If $h(\boldsymbol{x}_i) = P(y_i = 1|\boldsymbol{x}_i)$ then $\rightarrow P(y_i = 0|\boldsymbol{x}_i) = 1 - h(\boldsymbol{x}_i)$

Maximize the likelihood:

$$\begin{aligned}\mathcal{L} &= \prod_{\boldsymbol{x}_i|y_i=1} P(y_i = 1|\boldsymbol{x}_i) \prod_{\boldsymbol{x}_i|y_i=0} P(y_i = 0|\boldsymbol{x}_i) \\ &= \prod_i h(\boldsymbol{x}_i)^{y_i} \left(1 - h(\boldsymbol{x}_i)\right)^{1-y_i}\end{aligned}$$

$$\min_h - \log \mathcal{L} = \min_h \sum_i - \log\left(h(\boldsymbol{x}_i)\right) \cdot y_i - \log\left(1 - h(\boldsymbol{x}_i)\right) \cdot (1 - y_i)$$

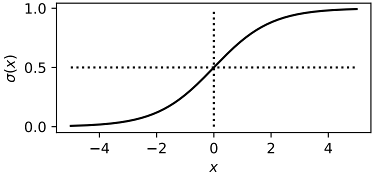


Choosing a suitable model

Problem: $0 < h(\boldsymbol{x}_i) < 1 \rightarrow$ naive linear model not suitable

Use sigmoid function instead:

$$\begin{aligned} \rightarrow h(\boldsymbol{x}) &= \sigma(\boldsymbol{x}\boldsymbol{w}) \\ &= \frac{1}{1 + e^{-\boldsymbol{x}\boldsymbol{w}}} \end{aligned}$$

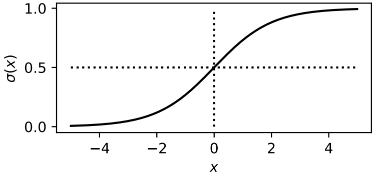


Choosing a suitable model

Problem: $0 < h(\boldsymbol{x}_i) < 1 \rightarrow$ naive linear model not suitable

Use sigmoid function instead:

$$\rightarrow h(\boldsymbol{x}) = \sigma(\boldsymbol{x}\boldsymbol{w}) = \frac{1}{1+e^{-\boldsymbol{x}\boldsymbol{w}}}$$

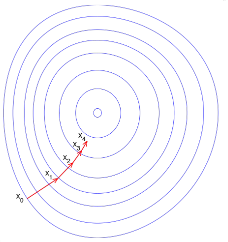


Insert our model into the loss function:

$$\begin{aligned}
\min_h -\log \mathcal{L} &= \min_h \sum_i -y_i \log\left(h(\boldsymbol{x}_i)\right) - (1-y_i)\log\left(1-h(\boldsymbol{x}_i)\right) \\
&= \min_{\boldsymbol{w}} \sum_i -y_i \log\left(\frac{1}{1+e^{-\boldsymbol{x}\boldsymbol{w}}}\right) - (1-y_i)\log\left(\frac{e^{-\boldsymbol{x}\boldsymbol{w}}}{1+e^{-\boldsymbol{x}\boldsymbol{w}}}\right) \\
&= \min_{\boldsymbol{w}} \sum_i -y_i\boldsymbol{x}\boldsymbol{w} + \log(1+e^{\boldsymbol{x}\boldsymbol{w}}) \rightarrow \text{can't be solved analytically!}
\end{aligned}$$

How can we minimize the loss function numerically?

$\rightarrow$ Common technique: Gradient descent



$$\boldsymbol{w}_n \rightarrow \boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \alpha \nabla \mathcal{L}(\boldsymbol{w}_n)$$

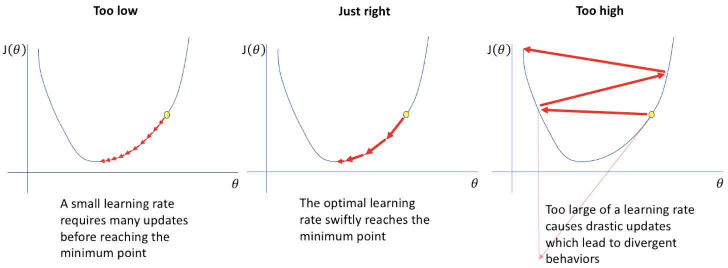
$\nearrow$

learning rate α

close to minimum $\rightarrow$ small gradient $\rightarrow$ small weight updates

Sensitivity to learning rate

Learning rate is one of the most critical parameters to tune



https://www.jeremyjordan.me/nn-learning-rate/

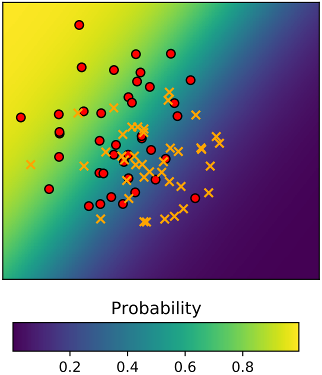
$\rightarrow$ Try behaviour for different orders of magnitude eg. $10^{-1} \ldots 10^{-6}$

Result

Gradient descent for logistic regression:

$$\nabla_{\boldsymbol{w}} - \log \mathcal{L}(\boldsymbol{w})$$
$$= \sum_i \left(y_i + \frac{1}{1 + e^{-\boldsymbol{x}_i \boldsymbol{w}}} \right) \boldsymbol{x}_i$$

$\rightarrow$ See exercise



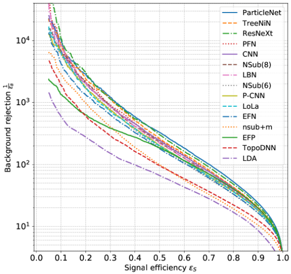
Evaluation

Receiver Operating Characteristic (ROC) curve:

	$h(\boldsymbol{x}) > D$	$h(\boldsymbol{x}) < D$	
top jet	true positive	false negative	TP + FN = 1
QCD jet	false positive	true negative	FP + TN = 1

Signal efficiency ϵ_S
$\sim$ true positive
$\rightarrow$ maximize

Background rejection $\dfrac{1}{\epsilon_B}$
$\sim$ true negative
$\rightarrow$ maximize



top vs QCD jets

Instructions to access exercises

1. (create a google account)
2. log in to your google account
3. https://colab.research.google.com/
4. a yellow window pops up → select GitHub
5. search for 'abutter'
6. select exercise 'Linear Regression'
7. select exercise 'Logistic Regression'