

Lezione 1

Chi siamo

- ▶ Luca Bottero mail: luca.bottero192@edu.unito.it
- ▶ Elia Cellini mail: elia.cellini@unito.it

Programma

- ▶ **Lezione 1:** Introduzione al machine learning
- ▶ **Lezione 2:** Regressione Lineare
- ▶ **Lezione 3:** Regressione Logistica
- ▶ **Lezione 4:** Explore SKlearn and Kaggle, SVC, Random Forest e Gradient Boosting
- ▶ **Lezione 5:** Unsupervised learning classico
- ▶ **Lezione 6:** Reti Neurali
- ▶ **Lezione 7:** Reti Neurali
- ▶ **Lezione 8:** Deep learning
- ▶ **Lezione 9/10:** TBA (Explore machine/deep learning)

Referenze

- ▶ Libri:
 - ▶ [Learning from Data: A Short Course; Abu-Mostafa and Magdon-Ismail; 2012]
 - ▶ [Pattern Recognition and Machine Learning; Bishop; 2006]
 - ▶ [The Elements of Statistical Learning; Hastie et al.; 2001]
 - ▶ [Deep Learning; Goodfellow, Bengio and Courville; 2015] [Approaching (Almost) Any Machine Learning Problem; Abhishek Thakur; 2020]
- ▶ Articoli: [Deep Learning; LeCun, Bengio and Hinton; 2015]
- ▶ Review: [A high-bias, low-variance introduction to Machine Learning for physicists; Mehta et al.; 2018]
- ▶ Medium/Wikipedia: Solo se sapete già qualcosa dell'argomento

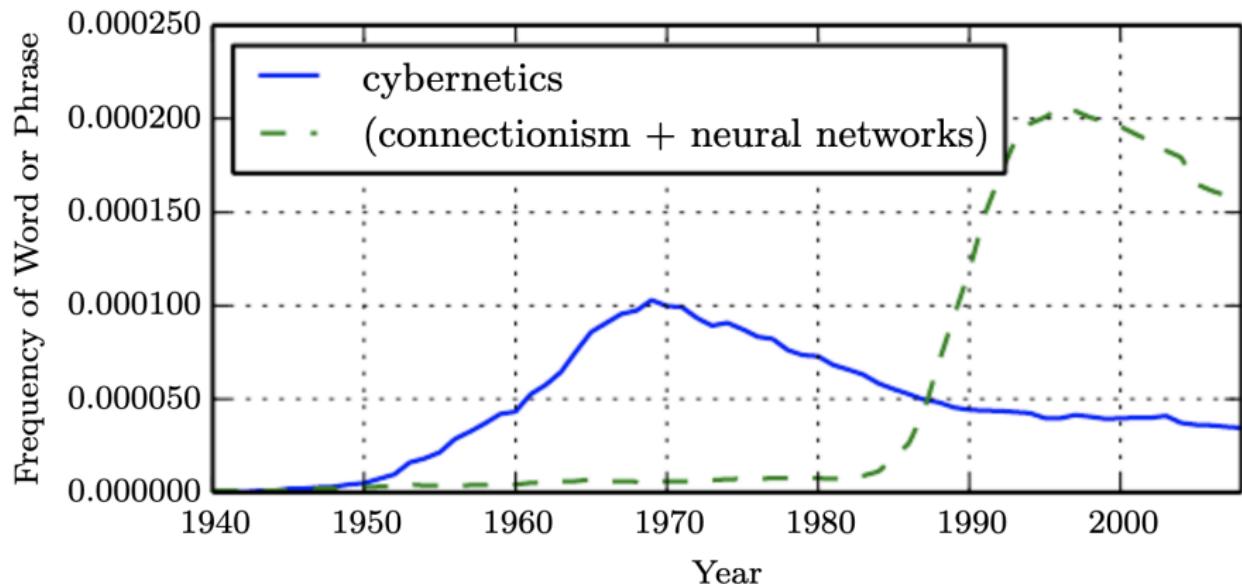
Strumenti

- ▶ Python
- ▶ Google Colab
- ▶ Google

Introduzione

Onde del machine learning

[Deep Learning; Goodfellow et al.]

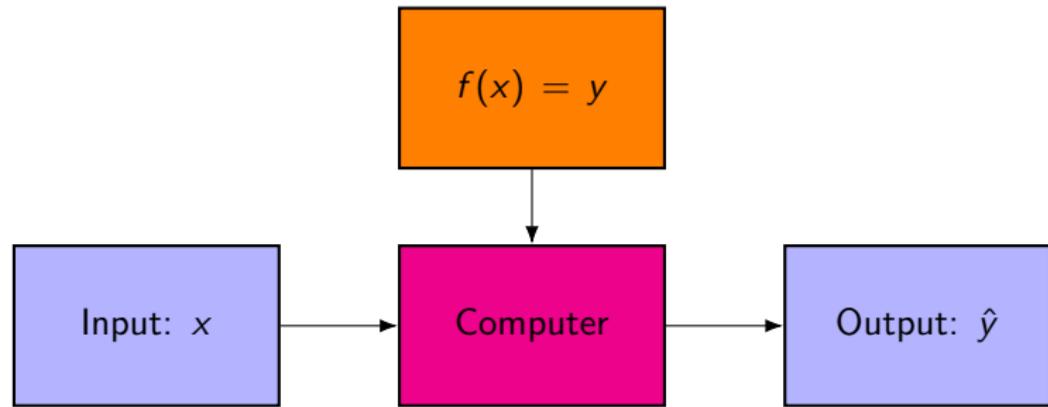


Terza ondata: Deep Learning

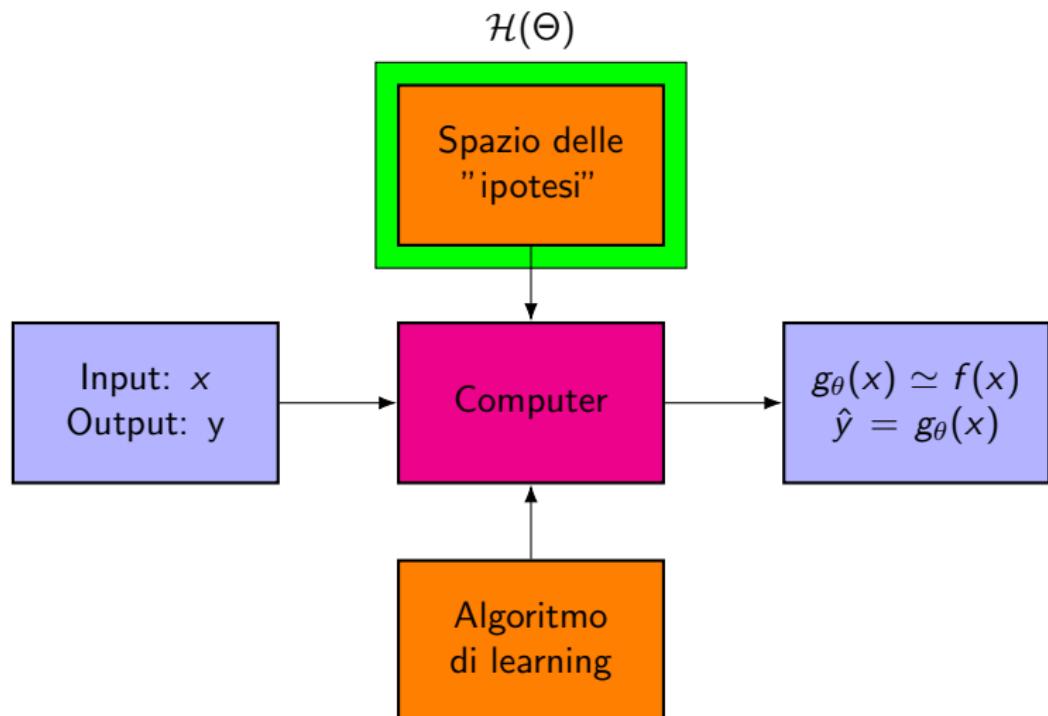
La terza ondata, comunemente associata al "deep learning", inizia nel 2006 da un articolo di Geoffrey Hinton. Nel 2018 viene assegnata la medaglia Turing a Yann LeCun, Geoffrey Hinton e Yoshua Bengio.



Programmazione classica



Machine Learning



Spazio delle ipotesi

Lo spazio delle ipotesi è generalmente costituito da un'insieme di fuzioni $g_\theta \in \mathcal{H}(\Theta)$:

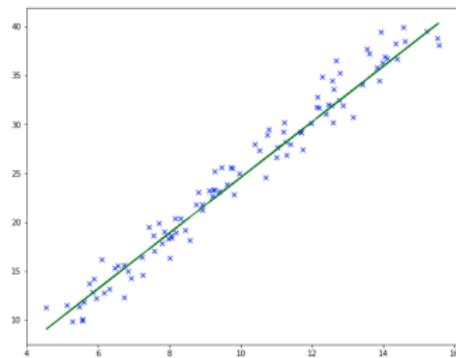
$$\hat{y} = g_\theta(x)$$

$$x \in D$$

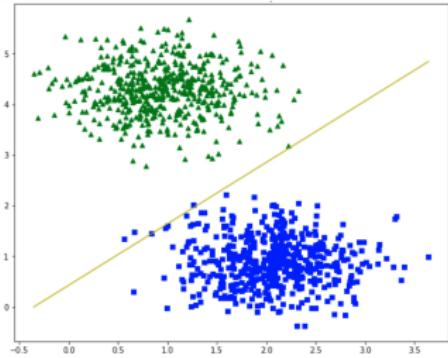
dove $x \in \mathbb{R}^d$ è un dato estratto da un dataset D costituito da N campioni con d features, y rappresenta l'output desiderato. Es:

$$\hat{y} = ax + b$$

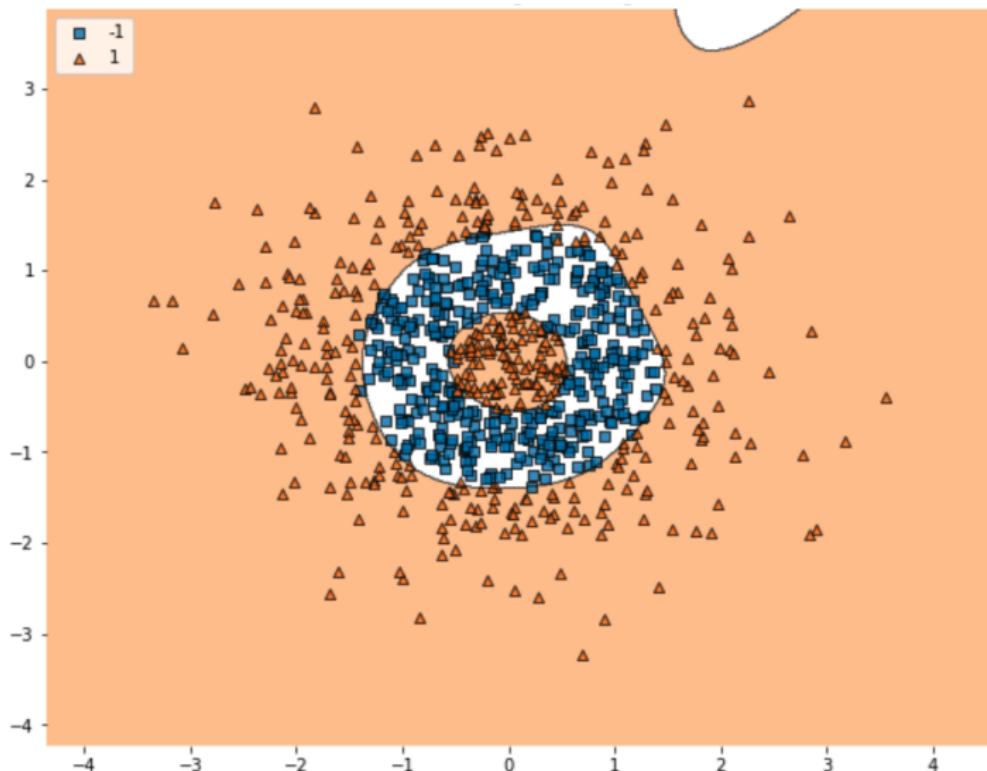
Regressione



Classificazione



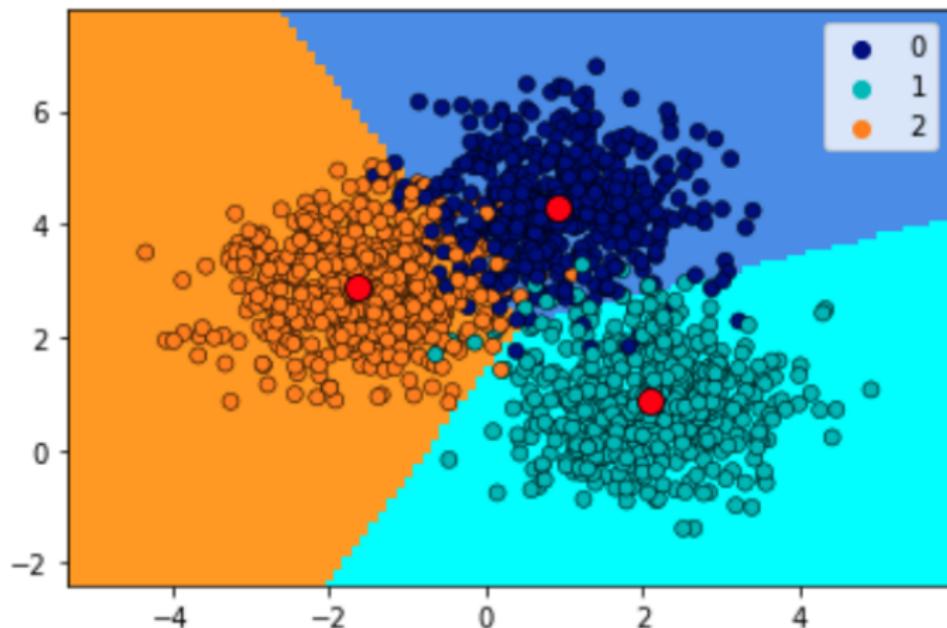
Modelli "complessi"



Dataset: Supervised and Unsupervised learning

Nel supervised learning, al processo di addestramento vengono forniti i dati x ed i corrispettivi target y .

Nell'unsupervised learning l'unico input sono i dati; gli output dei modelli possono essere, per esempio, delle classi y .

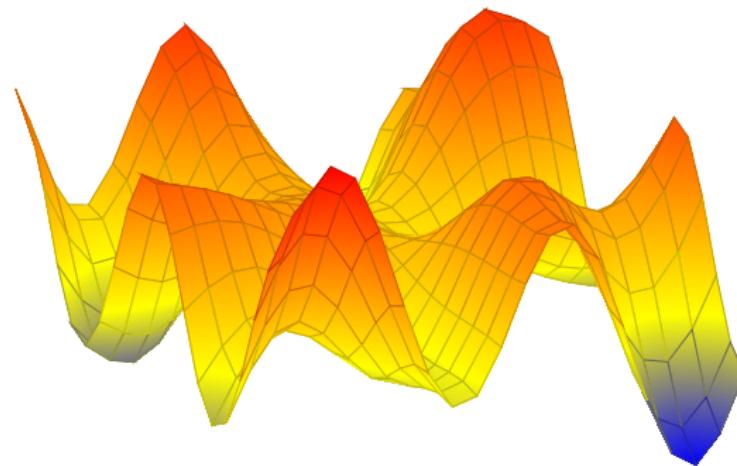


Addestramento

In generale, dato un dataset X ed un modello con parametri θ , l'ipotesi "migliore" è quella che minimizza un funzionale:

$$E(\theta) = L(X, g(\theta))$$

solitamente chiamato funzione di "loss" o "costo".



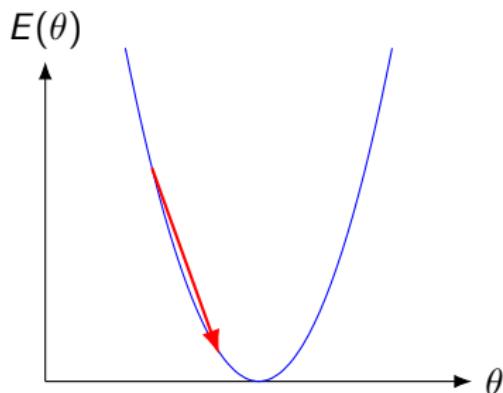
Algoritmo di learning

Un approccio molto popolare per stimare i minimi della loss si basa sull'utilizzo di metodi detti di "discesa stocastica del gradiente". I parametri del modello θ vengono esplorati tramite una procedura iterativa:

$$v_t = \eta_t \nabla_{\theta} E(\theta)$$

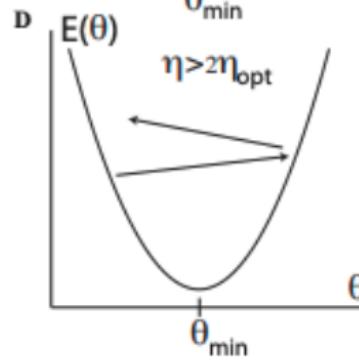
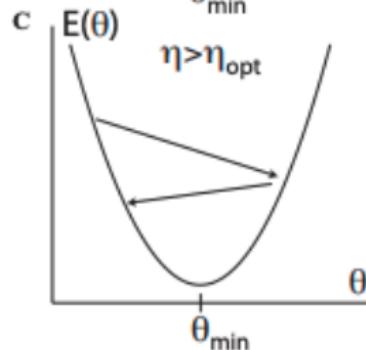
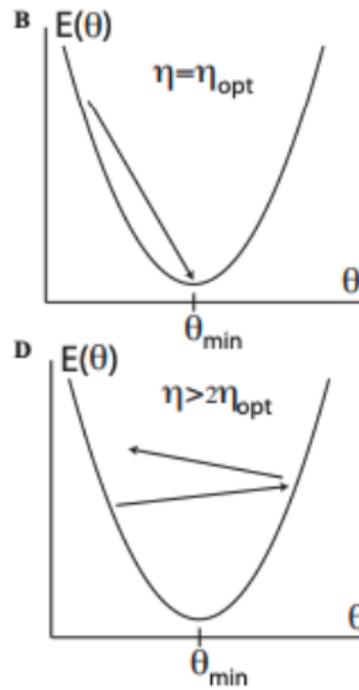
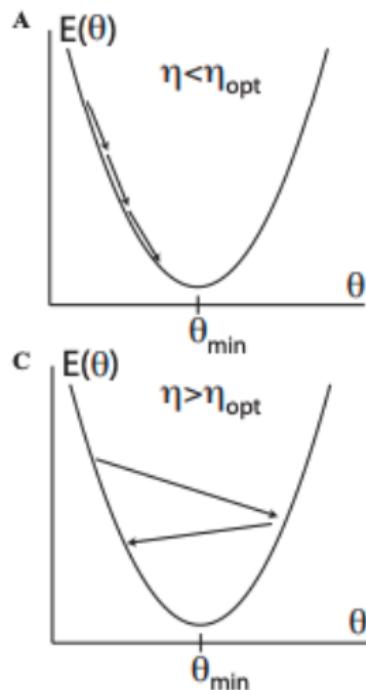
$$\theta_{t+1} = \theta_t - v_t$$

dove η è detto "learning rate" ed è un iperparametro fissato dal programmatore.



Learning rate

[A high-bias, low-variance introduction to Machine Learning for physicists; Mehta et al.; 2018]



Ingredienti del machine learning

- ▶ Spazio delle ipotesi g_θ
- ▶ Dataset (supervised learning, unsupervised learning, ecc)
- ▶ Ottimizzatori GD, ecc.. (iperparametri)

Errore: Tradeoff tra varianza e bias

Errore

L'errore di generalizzazione può essere riscritto come:

$$E_{out} = Bias^2 + Var + Noise$$

dove:

$$Bias^2 \propto (f(x) - \langle g_\theta(x) \rangle_D)^2$$

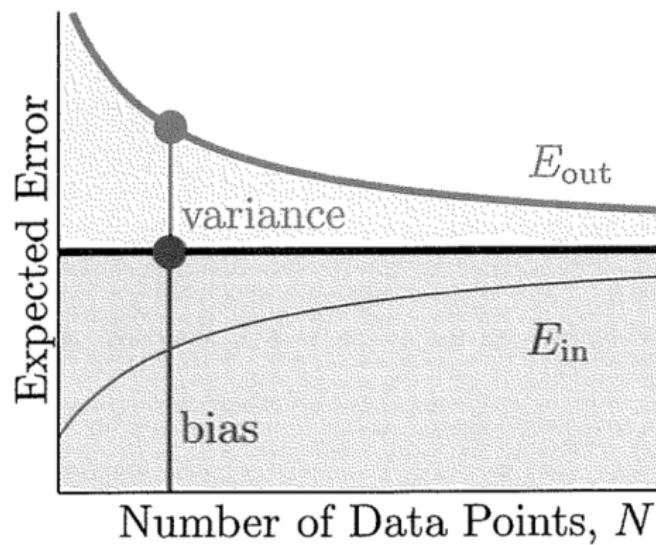
è l'errore del modello nel limite di dati infiniti mentre

$$Var \propto \langle (g_\theta(x) - \langle g_\theta(x) \rangle_D)^2 \rangle_D$$

è una stima delle fluttuazioni del modello dovuto al numero di campioni finiti.

Bias vs Variance

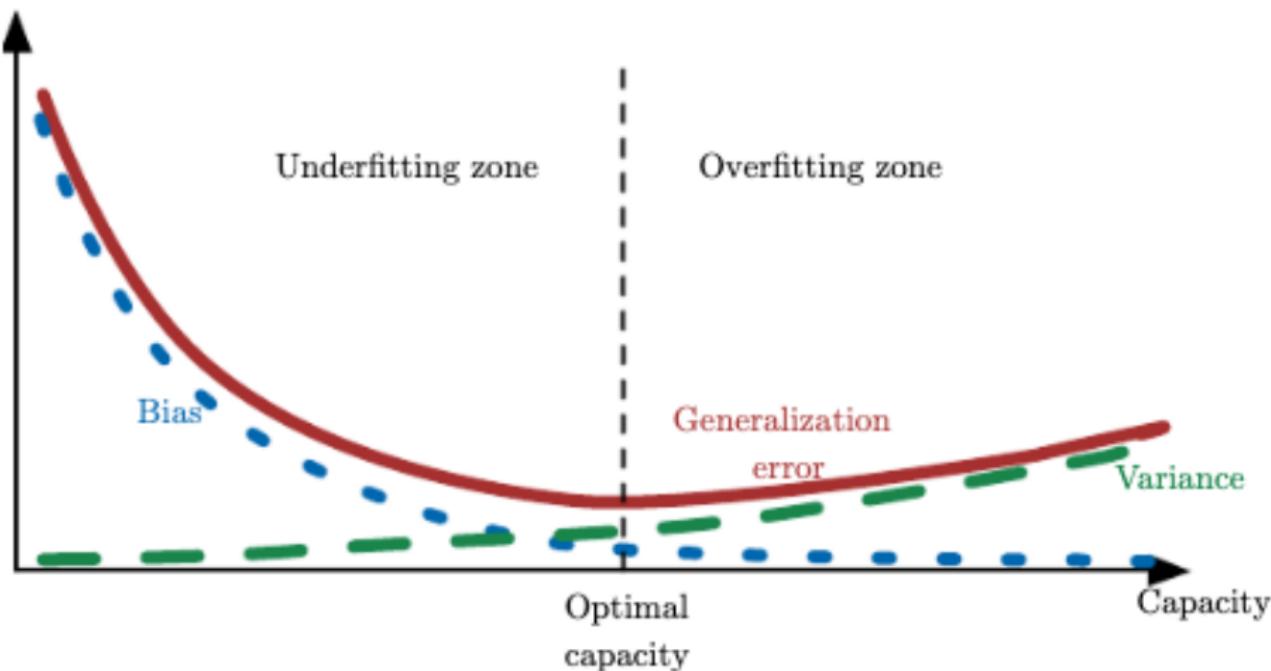
[Learning from Data: A Short Course; Abu-Mostafa and Magdon-Ismail; 2012]



Bias-Variance Analysis

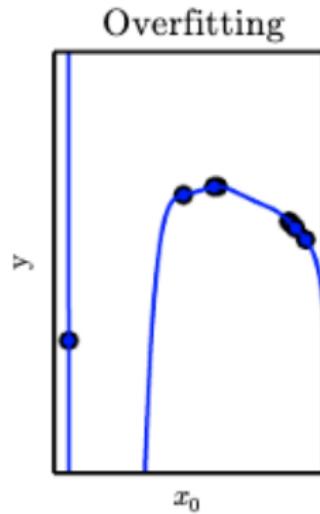
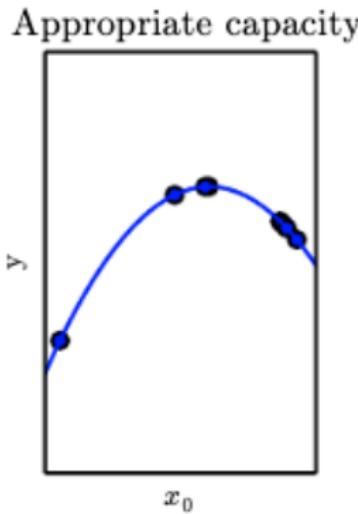
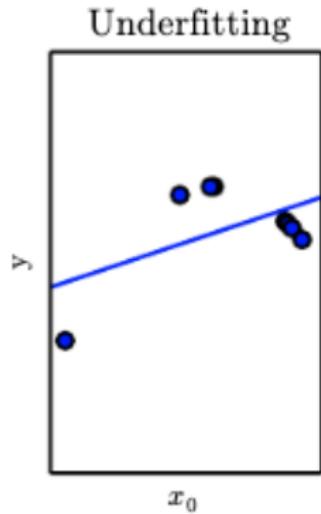
Error vs Capacity

[A high-bias, low-variance introduction to Machine Learning for physicists; Mehta et al.; 2018]



Underfitting vs Overfitting

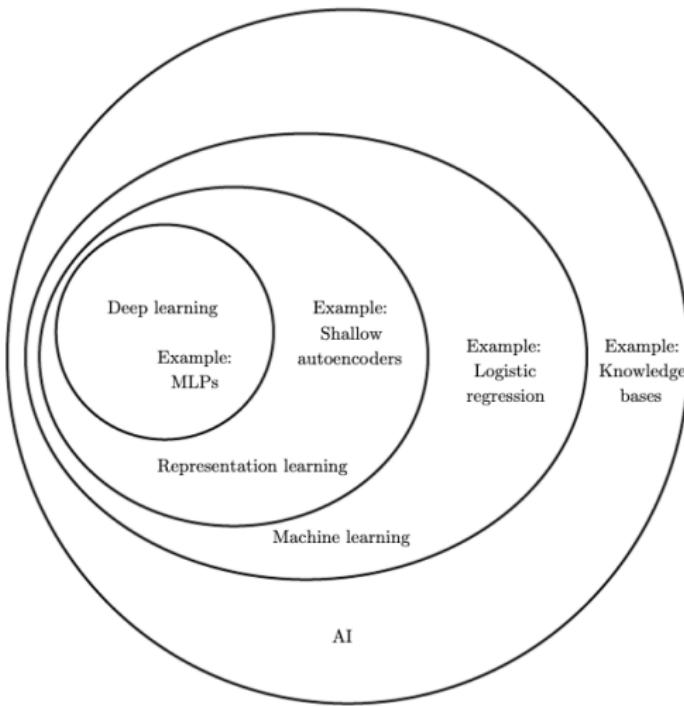
[A high-bias, low-variance introduction to Machine Learning for physicists; Mehta et al.; 2018]



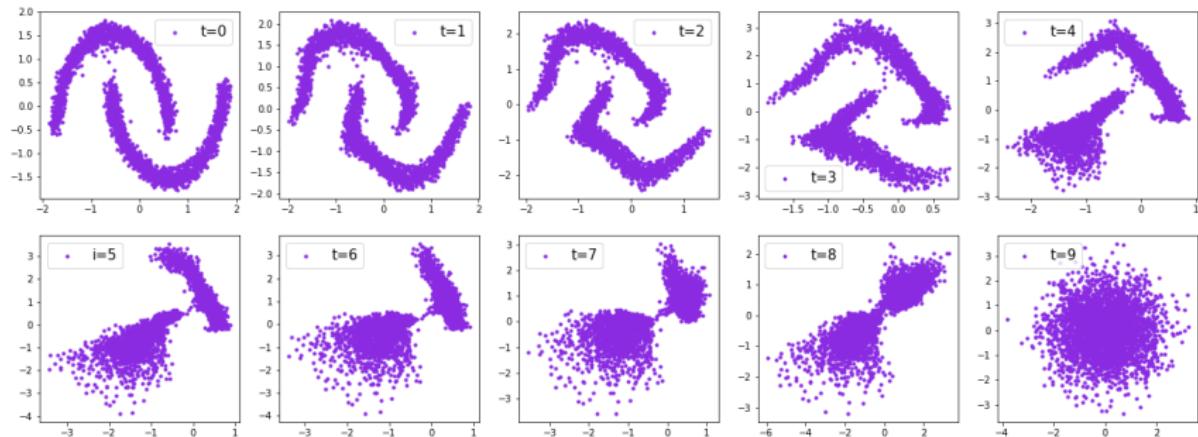
Deep learning

Deep Learning

[Deep Learning; Goodfellow, Bengio and Courville; 2015]

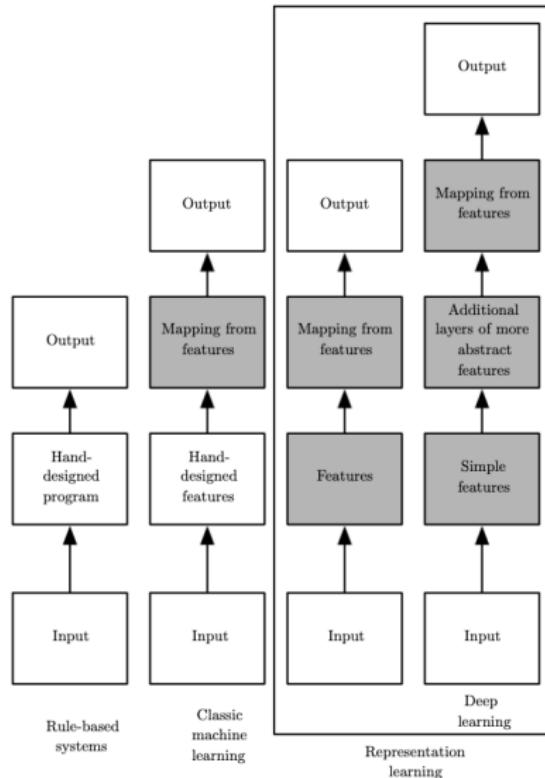


Representation Learning



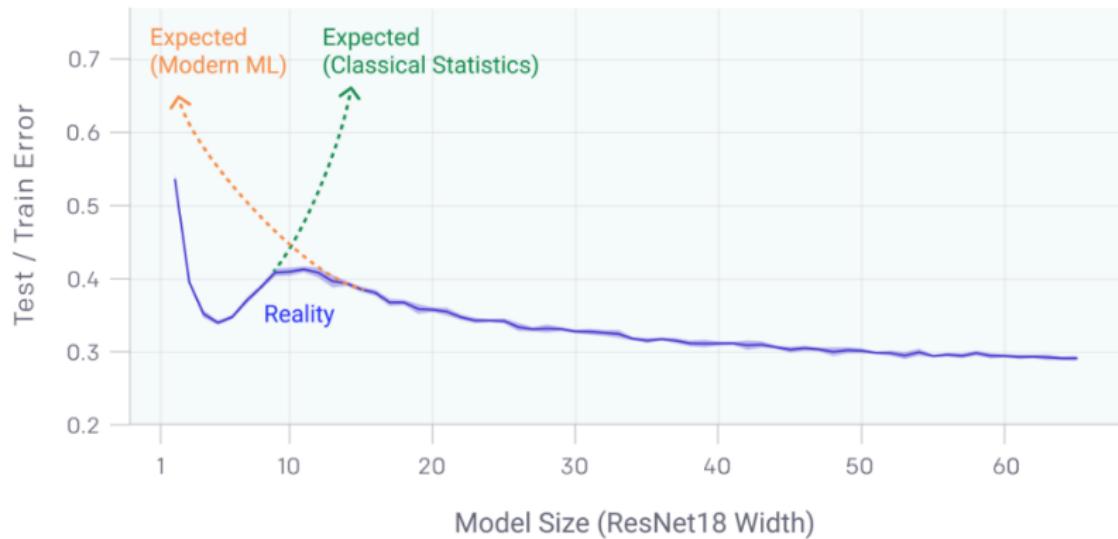
Features extraction

[Deep Learning; Goodfellow, Bengio and Courville; 2015]

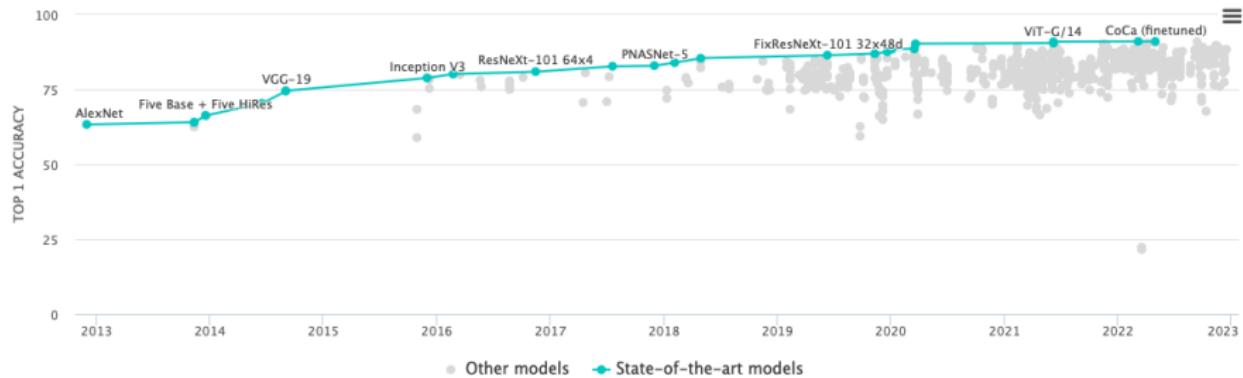


Double Descent

[Nakkiran et al.; arXiv:1912.02292]

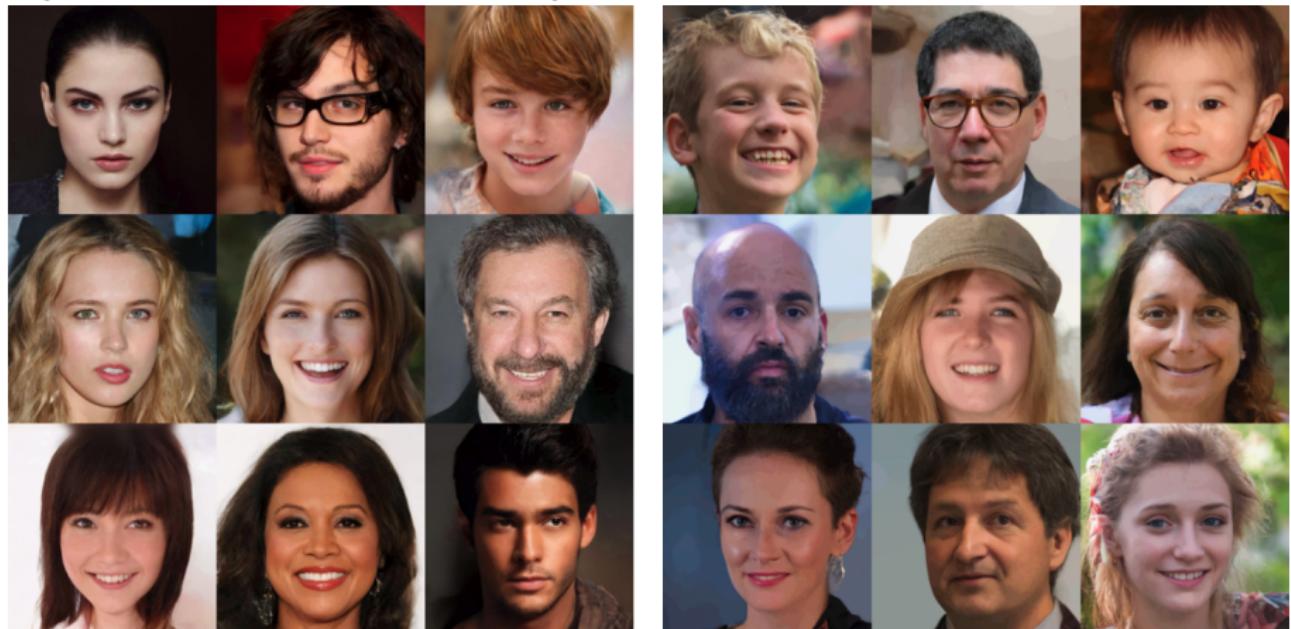


Imagenet

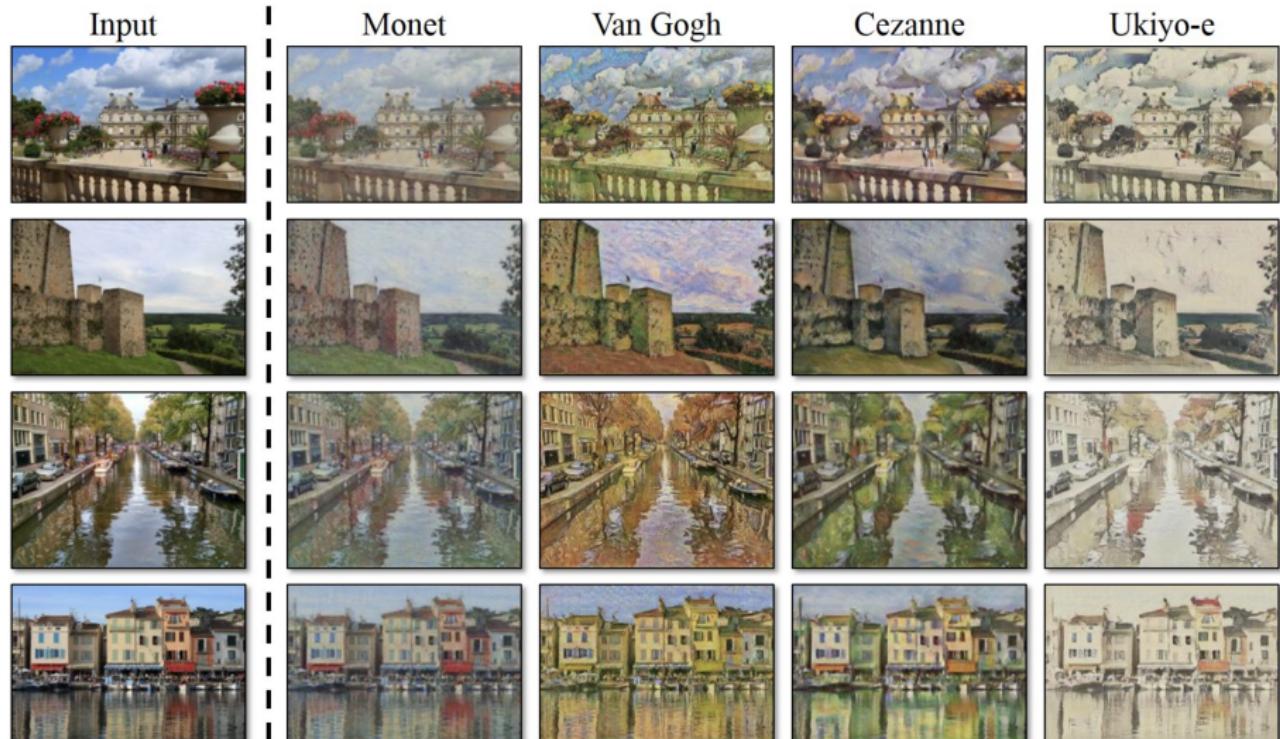


Generative models

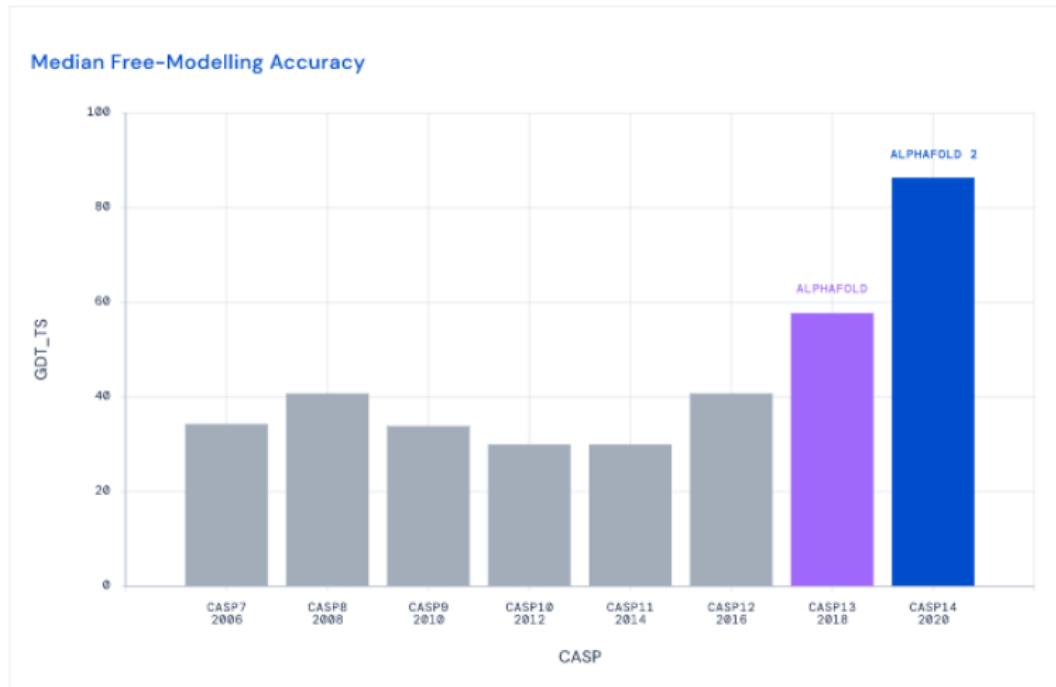
[arXiv:2203.13751; 2022]



CycleGAN



AlphaFold



Improvements in the median accuracy of predictions in the free modelling category for the best team in each CASP, measured as best-of-5 GDT.

ChatGPT3

E Scrivi una poesia per presentare chatGPT3



ChatGPT3 è una nuova intelligenza
Creata per parlare e rispondere
Con la sua mente artificiale
Può risolvere qualsiasi problema

Con la sua intelligenza artificiale
Può comprendere il linguaggio umano
E rispondere in modo preciso
Senza esitazioni o errori

ChatGPT3 è un'amica fedele
Sempre pronta ad aiutare
Con le sue risposte sagge
Può risolvere ogni tuo dubbio

Non esitare a chiedere aiuto
A questa amica digitale
Lei sarà sempre lì per te
Pronta ad ascoltare e rispondere

ChatGPT3, la tua nuova intelligenza
Sempre al tuo fianco, fedele e pronta
A risolvere ogni tuo problema
Con la sua mente artificiale.

E Scrivi una rete neurale convoluzionale in pytorch



Ecco come potresti scrivere una rete neurale convoluzionale di base in PyTorch:

```
import torch
import torch.nn as nn

class ConvNet(nn.Module):
    def __init__(self):
        super(ConvNet, self).__init__()
        self.conv1 = nn.Conv2d(3, 32, 3)
        self.conv2 = nn.Conv2d(32, 64, 3)
        self.pool = nn.MaxPool2d(2, 2)
        self.fc1 = nn.Linear(64 * 6 * 6, 128)
        self.fc2 = nn.Linear(128, 10)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = x.view(-1, 64 * 6 * 6)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x

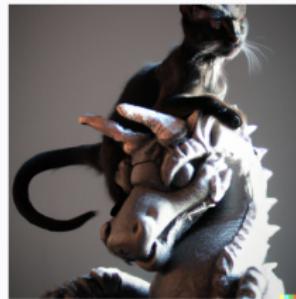
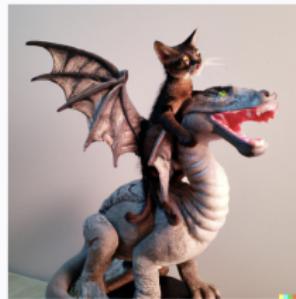
model = ConvNet()
```

La rete neurale consiste in due convolutional layers, seguiti da due fully-connected layers.

La funzione di attivazione ReLU viene applicata ai risultati delle convolutional layers e ai risultati del primo fully-connected layer. La funzione "forward" definisce il flusso di dati attraverso la rete.

photograph of a cat riding a dragon

Generate



Domande?

E

Concludi la prima lezione del corso di machine learning e saluta gli studenti



Alla fine della prima lezione del corso di machine learning, abbiamo imparato cos'è il machine learning e come funziona. Abbiamo anche visto alcuni esempi di come viene utilizzato in diverse applicazioni e come possiamo utilizzare algoritmi di machine learning per prevedere risultati o classificare oggetti.



Spero che questa lezione vi sia stata utile e che siate entusiasti di continuare a imparare di più su questo affascinante argomento. Non vedo l'ora di continuare il nostro viaggio insieme e di vedere cosa riusciremo a realizzare con il machine learning.

Grazie per aver partecipato alla lezione e vi saluto, a presto!