Hayat Boyu Derin Öğrenme

Bilimler Köyü, 1. Gün

Çağatay Yıldız

24.07.2023

Kaynaklar

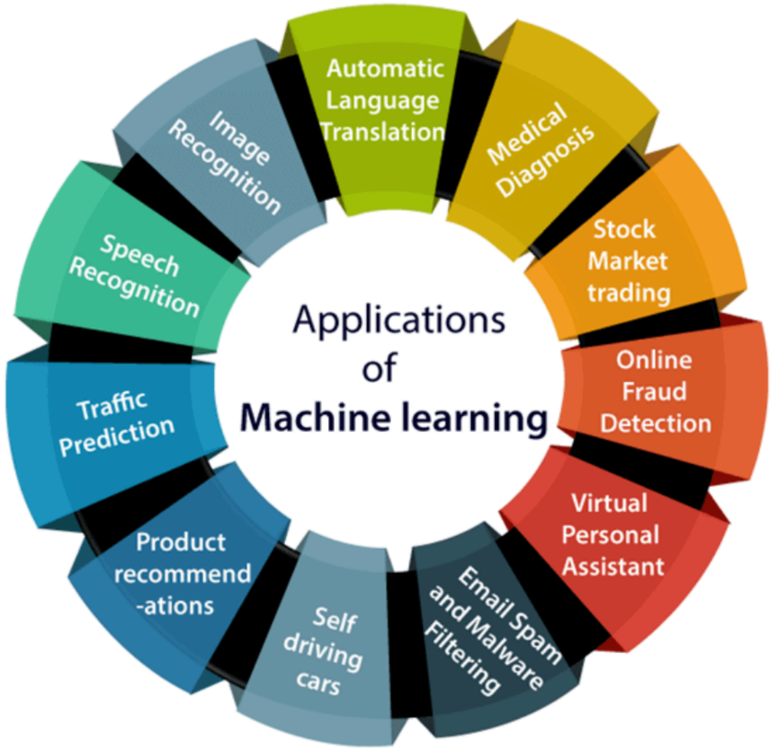
- Lifelong Machine Learning (Chen & Liu)
- Pattern Recognition and Machine Learning (Bishop)
- Deep Learning (Goodfellow et al)
- Mathematics for Machine Learning (Deisenroth et al)
- Tolga Birdal’in slaytlari, NMK Subat’23

İçerik

- Derin öğrenmenin temelleri
- Hayat boyu derin öğrenme
- Modüler öğrenme için grup teori
- Hayat boyu ayrışık öğrenme

Yapay öğrenme

- Matematiksel modeller olusturma ve bunları faydalı sonuclar verecekleri sekilde egitme



Son yıllardaki etkileyici uygulamalar

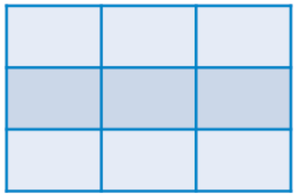
- Self-driving cars
- Large language models
- Protein folding
- Personal assistants

Neler değişti?

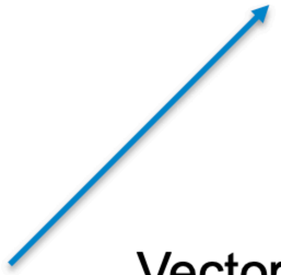
- Daha fazla data
- Daha büyük modeller
- Daha uygun inductive biases
- Daha çok compute

Machine learning terms

Data



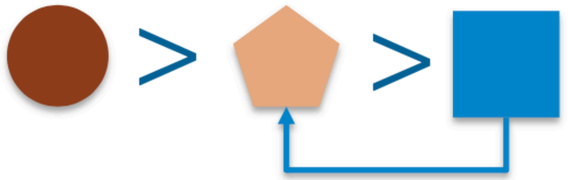
Features



Vectors

f

Model



Algorithm

Task
(Loss function)

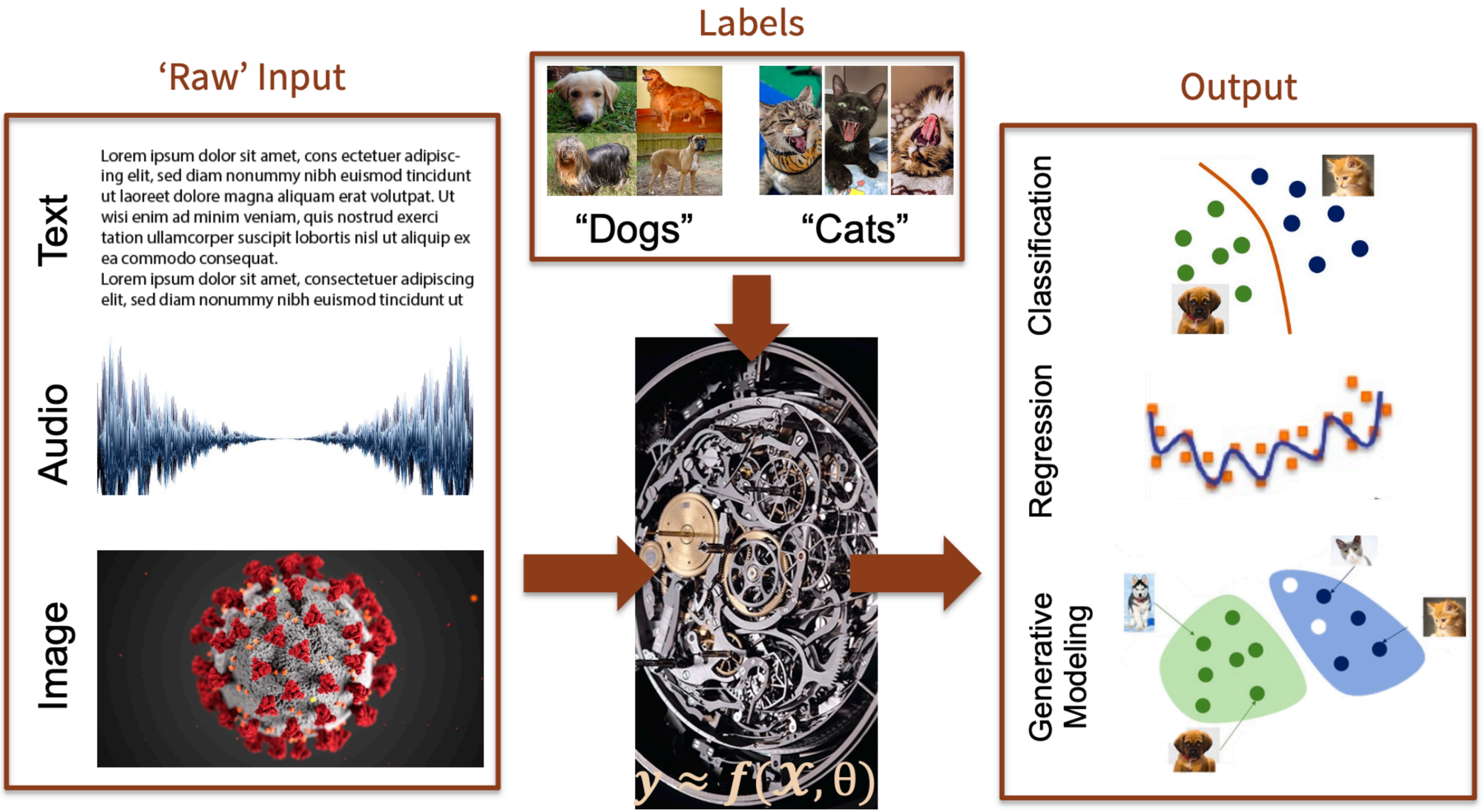
y

Labels
/ Targets

$y \approx f(X, \theta)$

Prediction

Machine learning

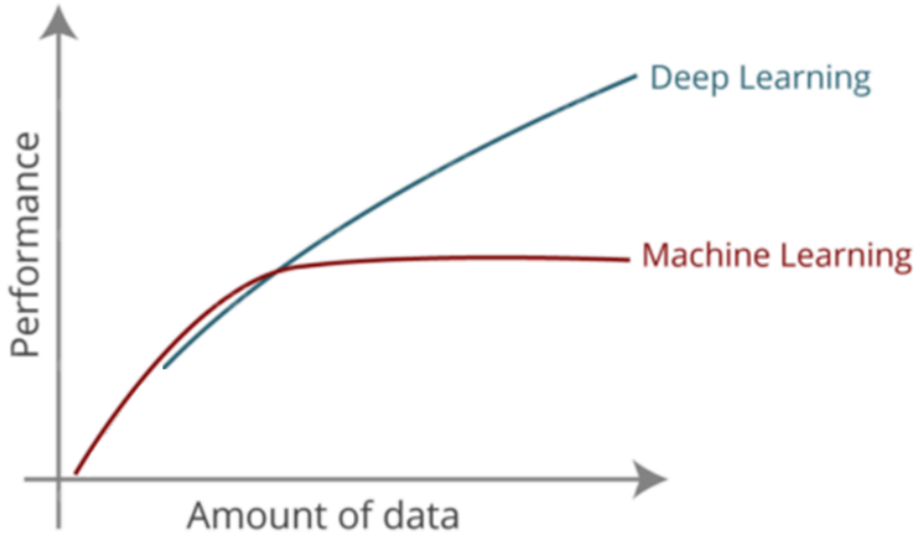


Machine learning

- Supervised (inductive) learning
  - Given: training data + desired outputs (labels)
- Unsupervised learning
  - Given: training data (without desired outputs)
- Semi-supervised learning
  - Given: training data + a few desired outputs
- Reinforcement learning
  - Rewards from sequence of actions
- Generative pre-training

  - We demonstrate that large gains on these tasks can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task.

Machine vs deep learning



A priori knowledge
Data selection
Data filtering & enhancing
Model selection
Learning technique choice
Experiment design
Avoiding brute force
Hybrid systems

Definitions

Vector

vector is a quantity possessing both magnitude and direction, represented by an arrow indicating the direction, and the length of which is proportional to the magnitude

Euclidean Space

$$\mathbb{R}^n = (x_1 \dots x_n) \; \forall x_i \in \mathbb{R}^1 = \mathbb{R}$$

$\mathbb{R}^1$: Line
$\mathbb{R}^2$: Plane
$\mathbb{R}^{3+}$: Space

$\mathbb{R}^n$ is a vector space:

$$\mathbf{x} + \mathbf{y} = (x_1 + y_1, \dots, x_n + y_n)$$

$$a\mathbf{x} = (ax_1, \dots, ax_2)$$

Hence, $\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^n$ are vectors.

Norm $\equiv$ *Length* of a vector:

$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$$

Matrix

Matrix is a rectangular array of real-valued scalars arranged in *m* horizontal rows and *n* vertical columns

Each element a_{ij} belongs to the i^{th} row and j^{th} column

The elements are denoted a_{ij} or $\mathbf{A}_{ij}$ or $[\mathbf{A}]_{ij}$ or $\mathbf{A}(\boldsymbol{i},\boldsymbol{j})$

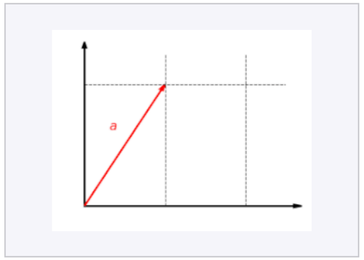
$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

For the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, the size (dimension) is $m \times n$ or (m, n)

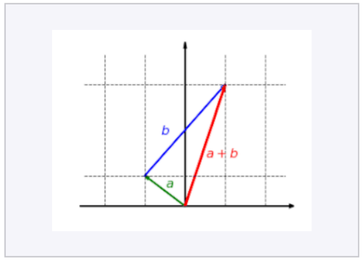
Matrices are denoted by bold-font upper-case letters

Matrix

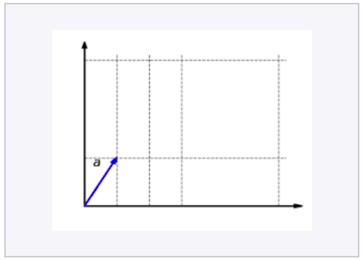
- Matrix addition, subtraction, scalar multiplication, matrix multiplication
- Transpose, square matrix, identity matrix, determinant



The function $f : \mathbb{R}^2 \to \mathbb{R}^2$ with $f(x, y) = (2x, y)$ is a linear map. This function scales the x component of a vector by the factor 2.



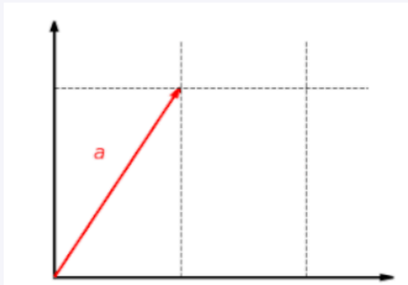
The function $f(x, y) = (2x, y)$ is additive: It does not matter whether vectors are first added and then mapped or whether they are mapped and finally added: $f(\mathbf{a} + \mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})$



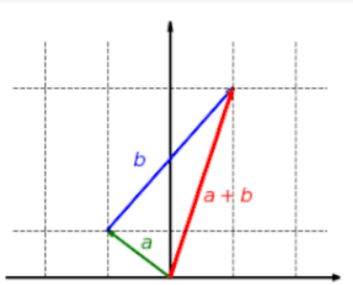
The function $f(x, y) = (2x, y)$ is homogeneous: It does not matter whether a vector is first scaled and then mapped or first mapped and then scaled: $f(\lambda\mathbf{a}) = \lambda f(\mathbf{a})$

Linear Maps

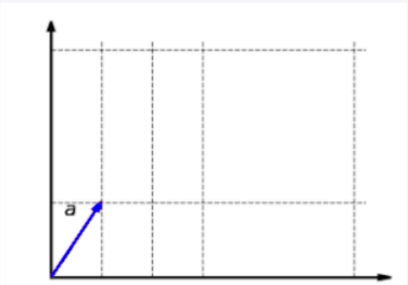
is a mapping between two vector spaces that preserves the operations of vector addition and scalar multiplication.



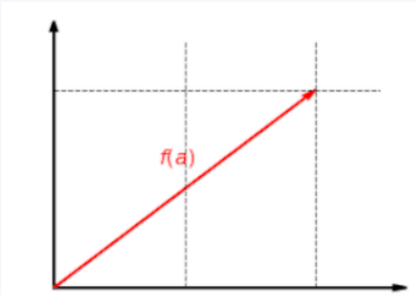
The function $f : \mathbb{R}^2 \to \mathbb{R}^2$ with $f(x, y) = (2x, y)$ is a linear map. This function scales the x component of a vector by the factor 2.

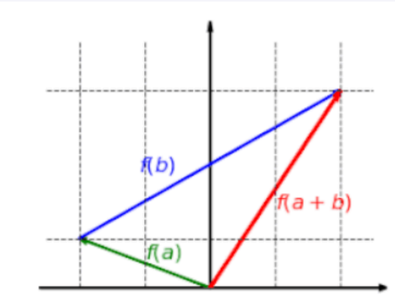


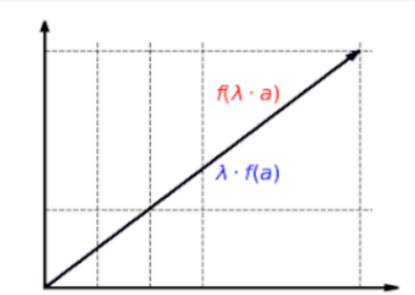
The function $f(x, y) = (2x, y)$ is additive: It does not matter whether vectors are first added and then mapped or whether they are mapped and finally added: $f(\mathbf{a} + \mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})$



The function $f(x, y) = (2x, y)$ is homogeneous: It does not matter whether a vector is first scaled and then mapped or first mapped and then scaled: $f(\lambda \mathbf{a}) = \lambda f(\mathbf{a})$







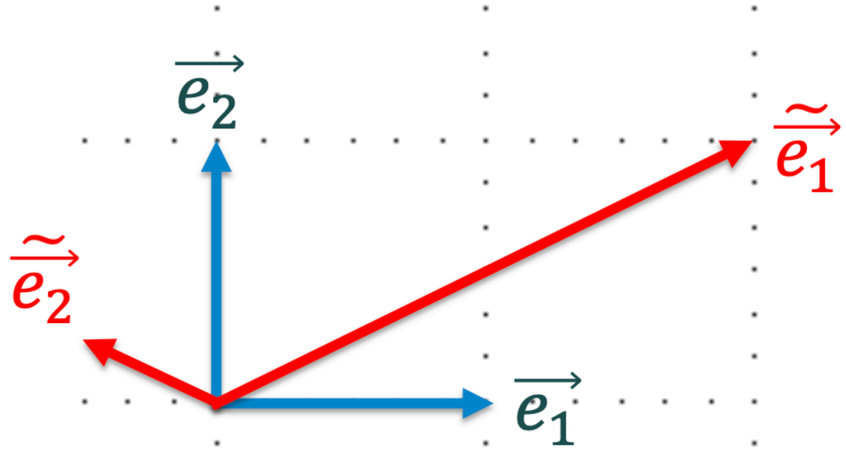
Basis transform

Old Basis: $\{\vec{e_1}, \vec{e_2}\}$

New Basis: $\{\vec{\tilde{e}_1}, \vec{\tilde{e}_2}\}$

$$\vec{\tilde{e}_1} = 2\,\vec{e_1} + 1\,\vec{e_2}$$
$$\vec{\tilde{e}_2} = -\tfrac{1}{2}\vec{e_1} + \tfrac{1}{4}\,\vec{e_2}$$

$$F = \begin{bmatrix} 2 & -\tfrac{1}{2} \\ 1 & \tfrac{1}{4} \end{bmatrix}$$



$$\vec{e_1} = \tfrac{1}{4}\,\vec{\tilde{e}_1} + (-1)\vec{\tilde{e}_2}$$
$$\vec{e_2} = \tfrac{1}{2}\,\vec{\tilde{e}_1} + 2\,\vec{\tilde{e}_2}$$

$$B = \begin{bmatrix} \tfrac{1}{4} & \tfrac{1}{2} \\ -1 & 2 \end{bmatrix}$$

Systems of linear equations

- To produce b1, we need a1i amounts of xi

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m \,, \end{aligned}$$

2.3.1 Particular and General Solution

Before discussing how to generally solve systems of linear equations, let us have a look at an example. Consider the system of equations

$$\begin{bmatrix} 1 & 0 & 8 & -4 \\ 0 & 1 & 2 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 42 \\ 8 \end{bmatrix}. \tag{2.38}$$

Functions

Functions & Continuity

$f : \mathbb{R}^n \mapsto \mathbb{R}^m$  $f(x) \in \mathbb{R}^m$ is defined only for $x \in \mathbb{R}^n$

range domain

Composition: For $f : A \mapsto \mathbb{R}^m$ and $g : B \mapsto \mathbb{R}^p$ where $B \subset \mathbb{R}^m$, $g \circ f(x) = g(f(x))$.

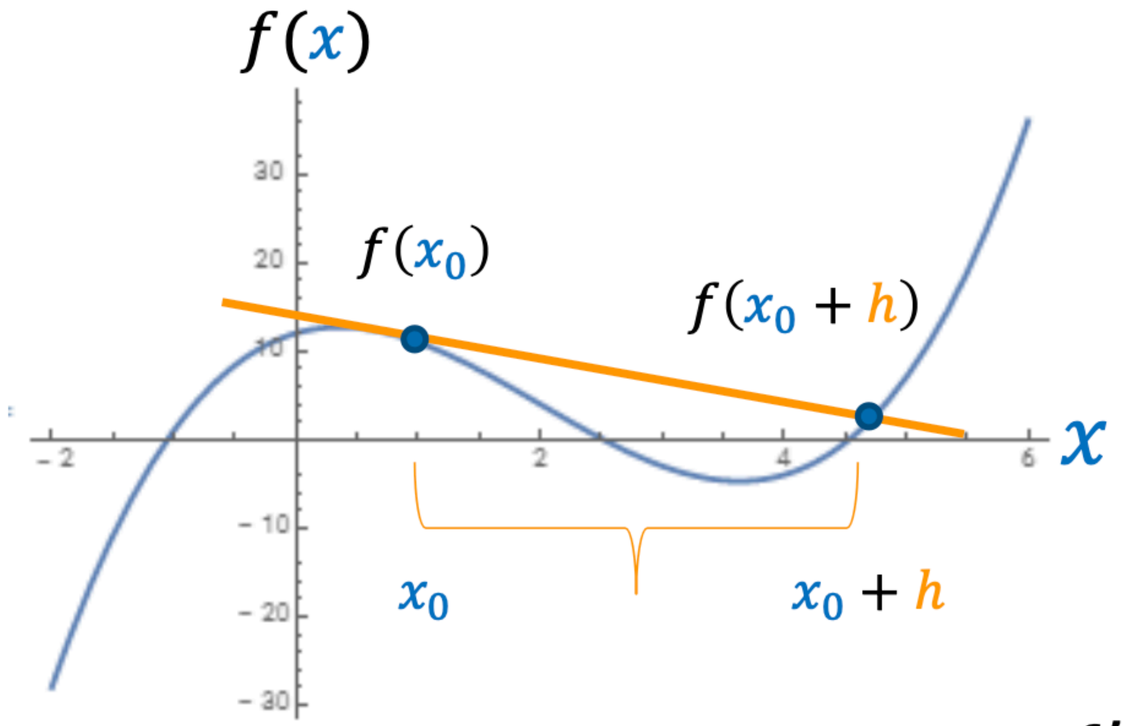
Domain of composition: The domain of $g \circ f = A \cup f^{-1}(B)$.

m-component function: $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x}))$

Projection function π_i: For an identity π, if $\pi(\mathbf{x}) = \mathbf{x}$ then $\pi_i(\mathbf{x}) = x_i$.

Continuity: $f : A \mapsto \mathbb{R}^m$ is **continuous at** $a \in A$ if $\lim_{x \to a} f(x) = f(a)$. f is continuous if it is continuous at each $a \in A$.

Derivatives



$$slope\ at\ x = f'(x) = \frac{df}{dx}$$

$$f(x) = x^3 - 6x^2 + 4x + 12$$

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

Derivatives

Power Rule:

$$\frac{d}{dx}x^n = nx^{n-1}$$

Exponential Rule:

$$\frac{d}{dx}e^x = e^x$$

Trig Rules:

$$\frac{d}{dx}\sin(x) = \cos(x)$$

$$\frac{d}{dx}\cos(x) = -\sin(x)$$

Sum Rule:

$$\frac{d}{dx}\big(f(x) + g(x)\big) = \frac{df}{dx} + \frac{dg}{dx}$$

Product Rule:

$$\frac{d}{dx}\big(f(x)g(x)\big) = \frac{df}{dx}g(x) + f(x)\frac{dg}{dx}$$

Chain Rule:

$$\frac{d}{dx}\Big(f\big(g(x)\big)\Big) = \frac{df}{dg}\frac{dg}{dx}$$

Chain rule

Chain Rule:

$$\frac{d}{dx}\left(f(g(x))\right) = \frac{df}{dg}\frac{dg}{dx}$$

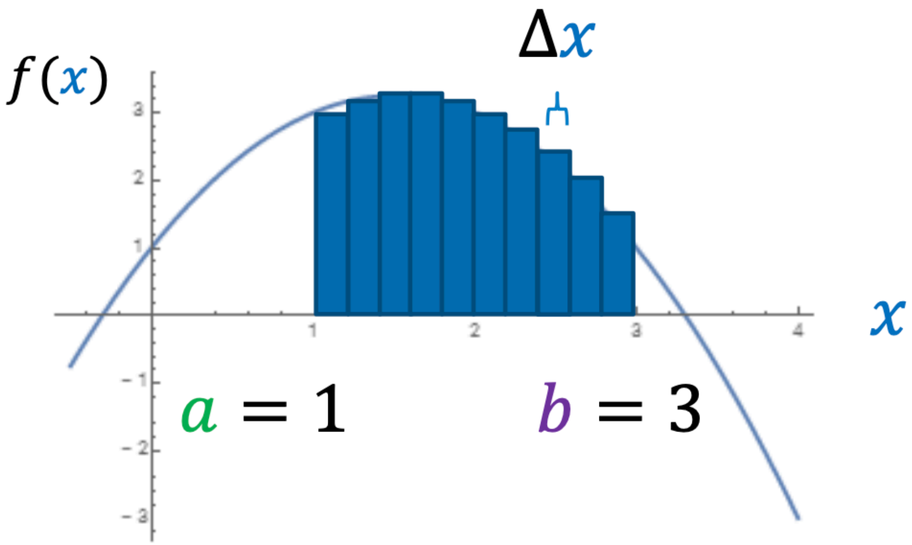
$$\frac{d}{dx}(\sin(2x))$$

$$= \frac{d}{dg}(\sin(g)) \cdot \frac{d}{dx}(2x)$$

$$= \cos(g) \cdot 2$$

$$= \cos(2x) \cdot 2$$

Integral

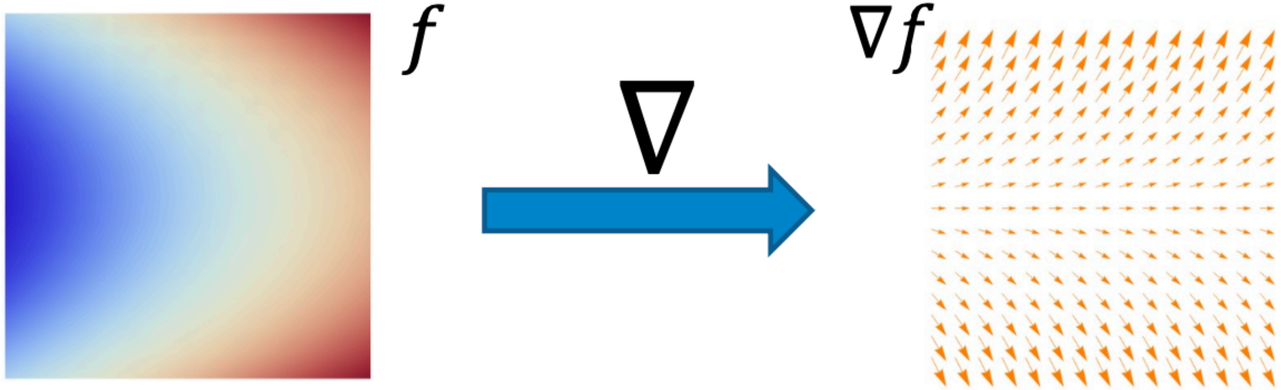


$$f(x) = -x^2 + 3x + 1$$

$$\int_a^b f(x)\,dx \equiv \lim_{\max \Delta x_k \to 0} \sum_{k=1}^{n} f(x_k^*)\,\Delta x_k$$

Gradient

- Gradient takes a function (i.e. scalar field) and produces a vector field
  - Vector points in direction of greatest increase
  - Vector magnitude is proportional to steepness (rate of increase)



$$f(x,y) = y^2 + x - ½$$

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix}$$

$$\nabla f = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} (y^2 + x - ½)$$

$$= \begin{bmatrix} \frac{\partial}{\partial x}(y^2 + x - ½) \\ \frac{\partial}{\partial y}(y^2 + x - ½) \end{bmatrix} = \begin{bmatrix} 1 \\ 2y \end{bmatrix}$$

Machine learning

Machine Learning: $f(x, w) \approx y \xrightarrow{\text{training}} w^\star \ (\textit{Optimization})$

Model hypothesis: f
Parameters: w
Data space: $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
Loss function: ℓ

$$w^\star = \arg\min_w \sum_{i=1}^{n} \ell\big(f(x_i, w), y_i\big)$$

Empirical Risk $\hat{\mathcal{R}}$ on n training samples $\mathcal{S} \in \mathcal{Z}^n$

$$\hat{\mathcal{R}}(w, S) := \frac{1}{n} \sum_{i=1}^{n} \ell(w, z_i)$$

Inductive biases

Inductive bias on the parameters

$$w^\star = \arg\min_w \sum_{i=1}^{n} \Big(\ell\big(f(x_i, w), y_i\big) + h(w) \Big)$$

data regularizer

Inductive bias on the hypotheses class

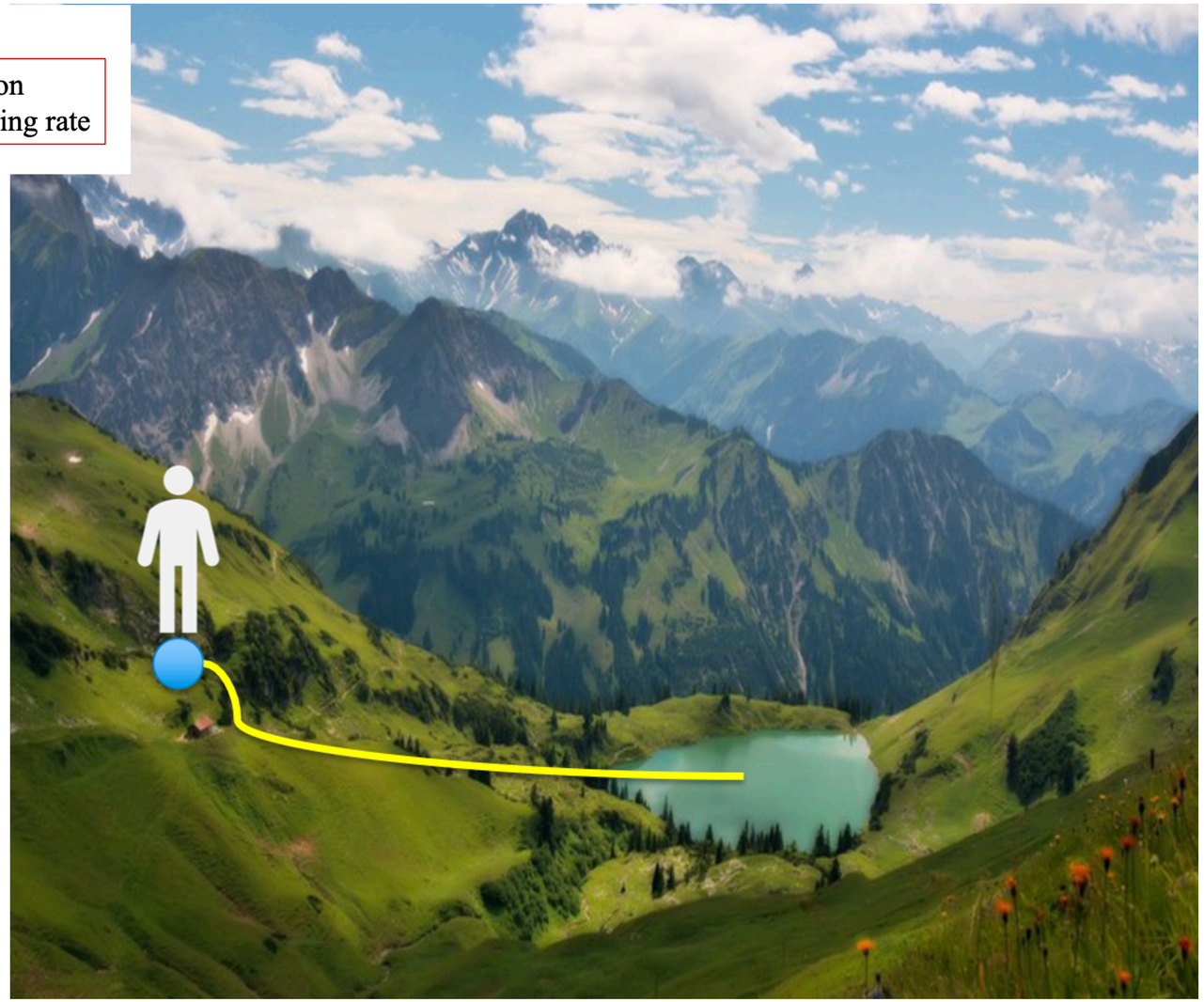
$$\hat{f} = \arg\min_{f \in \mathcal{F}_\delta} \hat{\mathcal{R}}(f)$$

Gradient descent

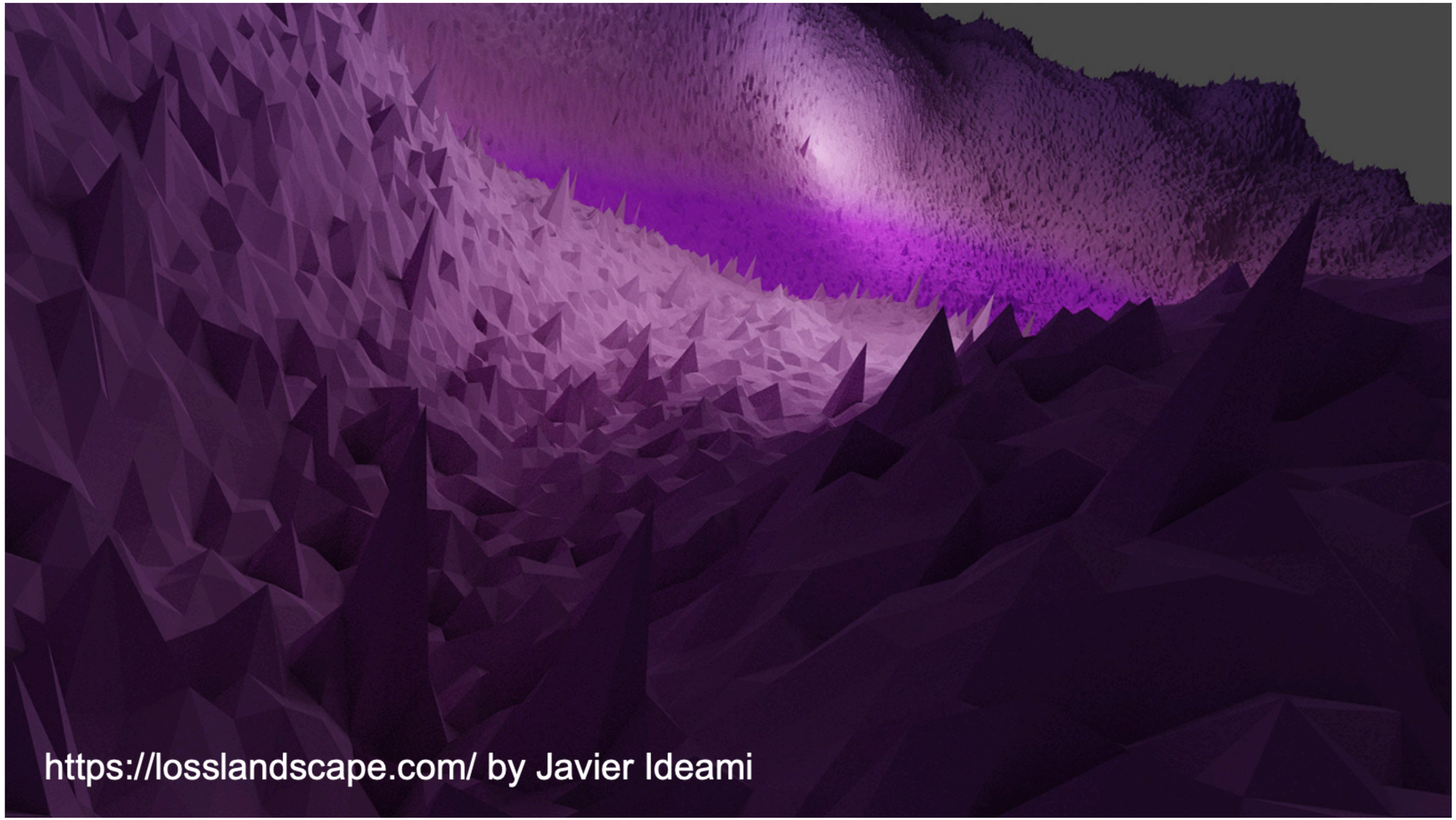
(Batch) Gradient Descent

$$\beta_{k+1} = \beta_k - \frac{\alpha_k}{n} \sum_{i=1}^{n} \nabla E_i(\beta_k)$$

k : iteration
$\propto_k$: learning rate

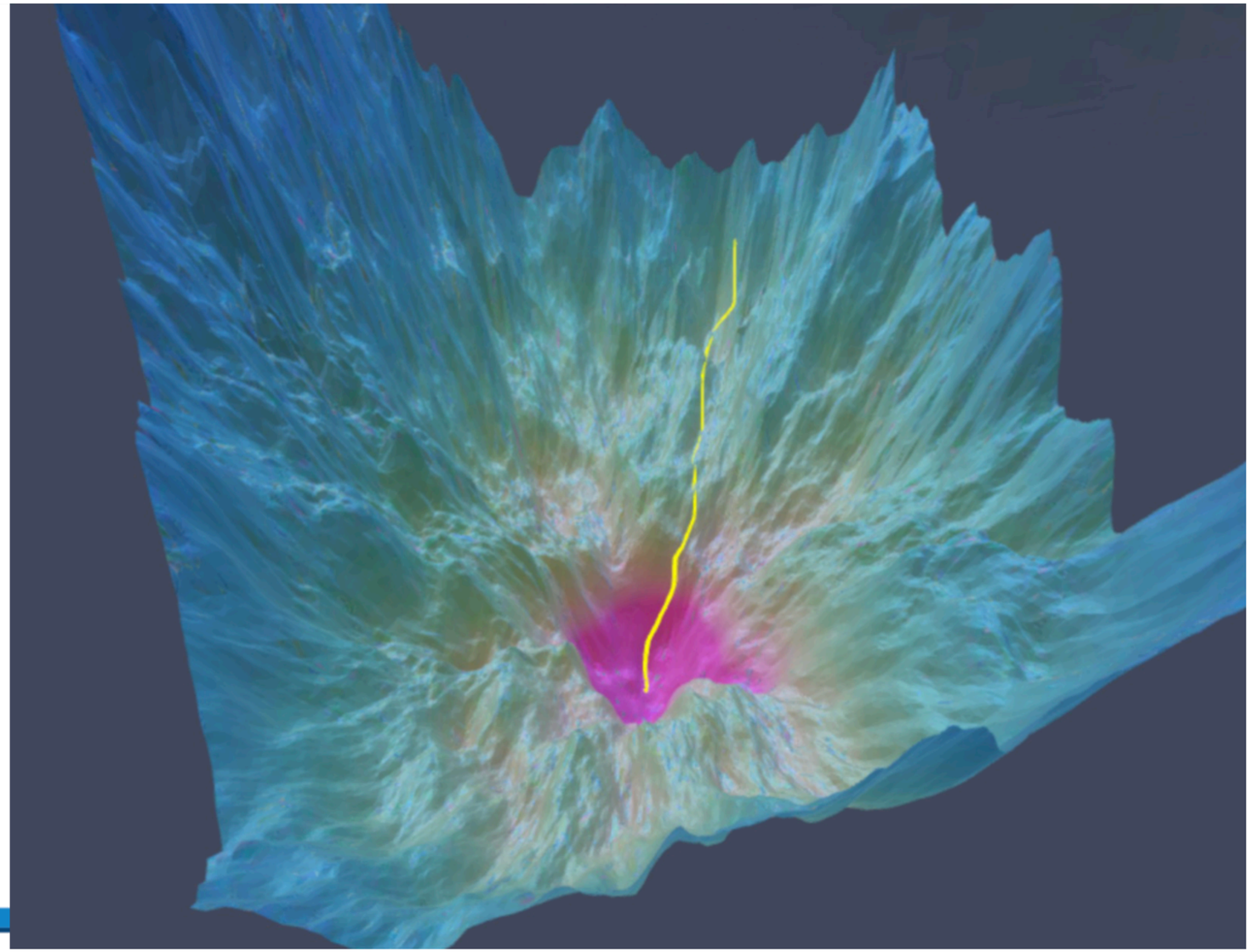


Gradient descent



Gradient descent

A Descent Trajectory



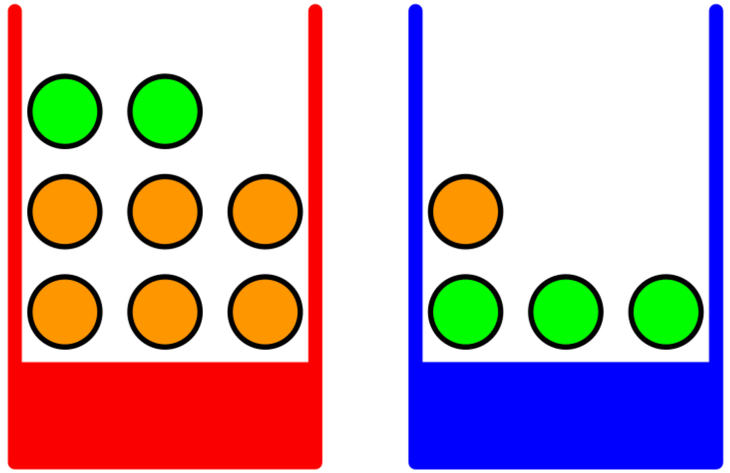
Taylor’s expansion

- Gradient descent: f(x+h) = f(x) +f’(x)h + o(h^2(

$$T_\infty(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

Probabilistic view

Figure 1.9 We use a simple example of two coloured boxes each containing fruit (apples shown in green and oranges shown in orange) to introduce the basic ideas of probability.



$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}.$$

$$p(X = x_i) = \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

Rules of prob

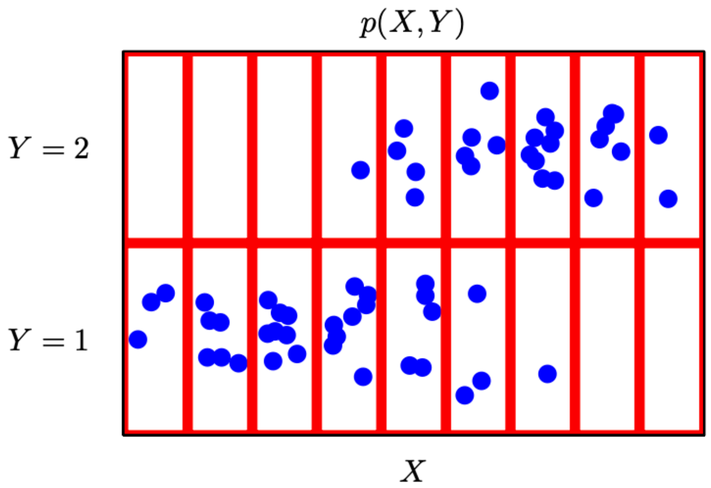
The Rules of Probability

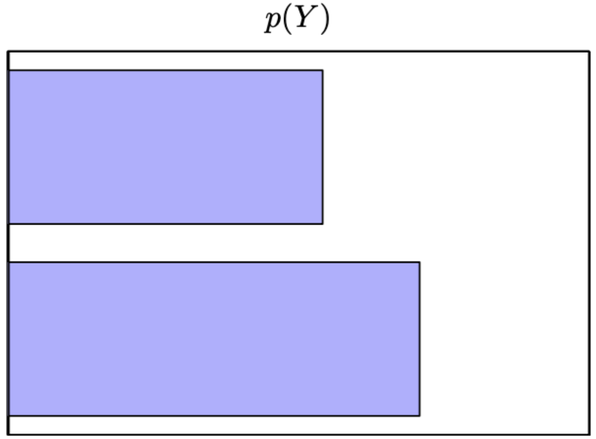
sum rule $$p(X) = \sum_{Y} p(X, Y)$$

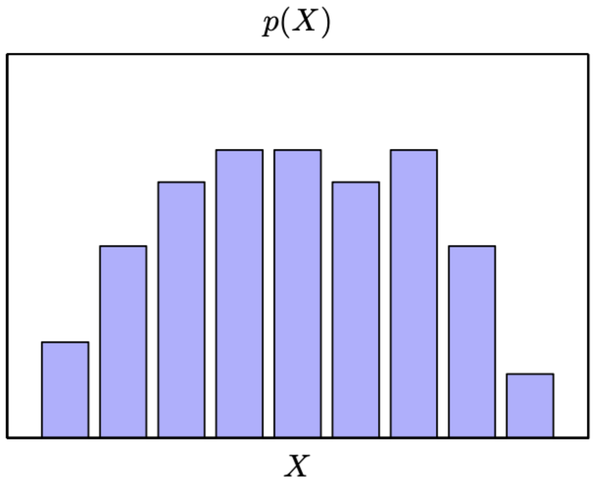
product rule $$p(X, Y) = p(Y|X)p(X).$$

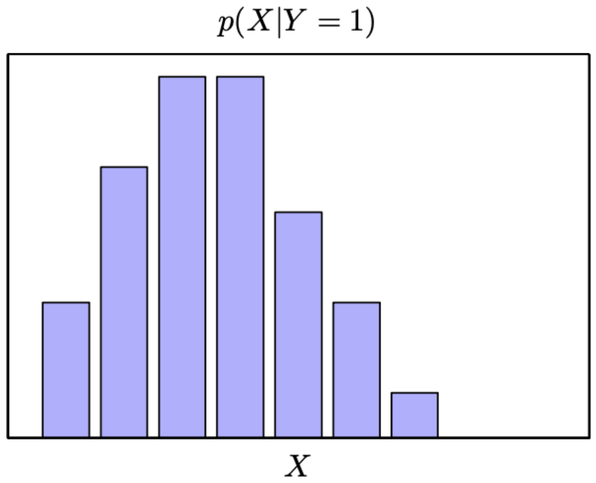
$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

Rules of probability







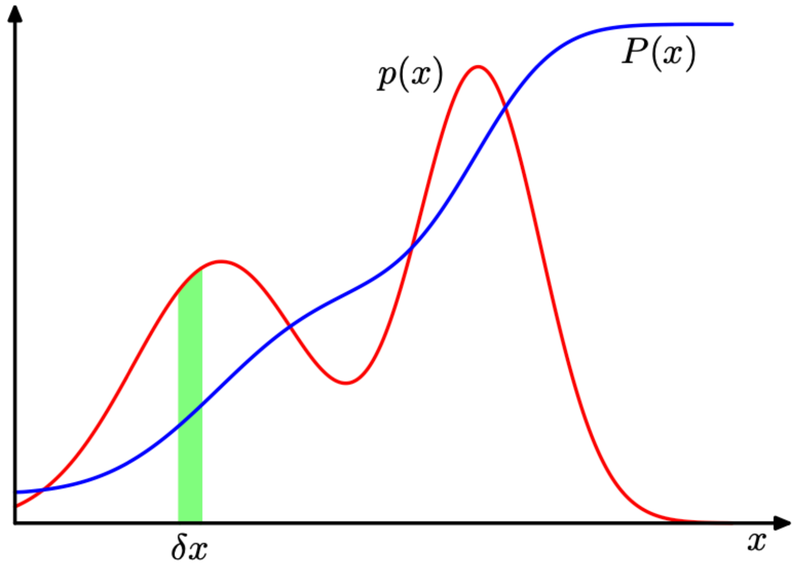


Probability densities

limit ourselves to a relatively informal discussion. If the probability of a real-valued variable x falling in the interval $(x, x + \delta x)$ is given by $p(x)\delta x$ for $\delta x \to 0$, then $p(x)$ is called the *probability density* over x. This is illustrated in Figure 1.12. The probability that x will lie in an interval (a, b) is then given by

$$p(x \in (a, b)) = \int_a^b p(x)\,\mathrm{d}x. \tag{1.24}$$

The concept of probability for discrete variables can be extended to that of a probability density $p(x)$ over a continuous variable x and is such that the probability of x lying in the interval $(x, x+\delta x)$ is given by $p(x)\delta x$ for $\delta x \to 0$. The probability density can be expressed as the derivative of a cumulative distribution function $P(x)$.



Because probabilities are nonnegative, and because the value of x must lie somewhere on the real axis, the probability density $p(x)$ must satisfy the two conditions

$$p(x) \geqslant 0 \tag{1.25}$$

$$\int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1. \tag{1.26}$$

Probability densities

$$p(\mathbf{x}) \geqslant 0$$
$$\int p(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1$$

$$p(x) = \int p(x,y)\,\mathrm{d}y$$
$$p(x,y) = p(y|x)p(x).$$

Expectations, mean and variance

$$\mathbb{E}[f] = \int p(x) f(x) \, \mathrm{d}x.$$

The *variance* of $f(x)$ is defined by

$$\mathrm{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] \tag{1.38}$$

and provides a measure of how much variability there is in $f(x)$ around its mean value $\mathbb{E}[f(x)]$. Expanding out the square, we see that the variance can also be written in terms of the expectations of $f(x)$ and $f(x)^2$

Bayesian ML

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

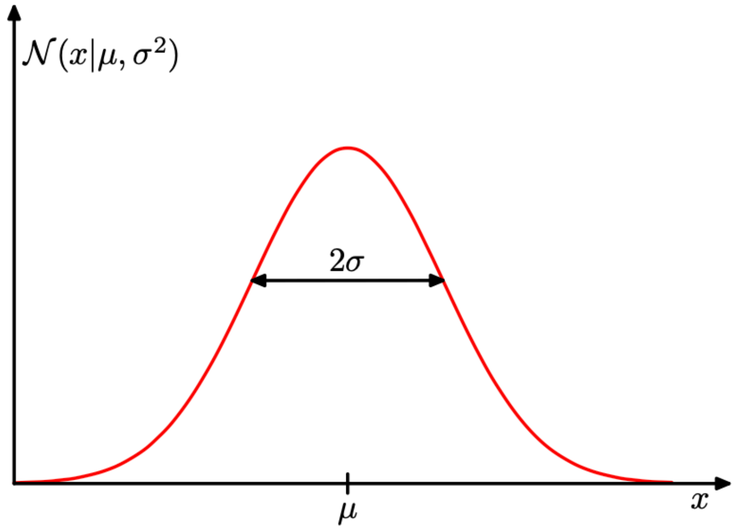
$$p(\mathcal{D}) = \int p(\mathcal{D}|\mathbf{w})p(\mathbf{w})\,\mathrm{d}\mathbf{w}.$$

Gaussian distribution

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

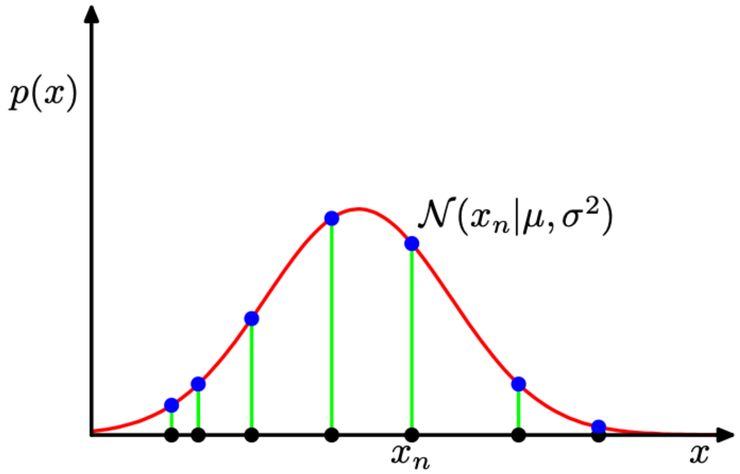
$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Plot of the univariate Gaussian showing the mean μ and the standard deviation σ.



Gaussian fitting

Illustration of the likelihood function for a Gaussian distribution, shown by the red curve. Here the black points denote a data set of values $\{x_n\}$, and the likelihood function given by (1.53) corresponds to the product of the blue values. Maximizing the likelihood involves adjusting the mean and variance of the Gaussian so as to maximize this product.



$$p(\mathbf{x}|\mu,\sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(x_n|\mu,\sigma^2\right).$$

$$\ln p\left(\mathbf{x}|\mu,\sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi).$$

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad \sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_{\mathrm{ML}})^2$$

Parameter fitting

- Switch to MoML, sec 9.1
  - Define inputs x, outputs y, likelihood, linear mapping
  - ML estimation
  - Feature space ==> overfitting
  - Prior, MAP ===> regularisation
  - Bayesian regression

	24 Temmuz - Pazartesi	25 Temmuz - Salı	26 Temmuz - Çarşamba	emmuz - Perşe	28 Temmuz - Cuma	29 Temmuz - Cumartesi
	Giris ve hayat boyu öğrenme	**Optimizasyona giris**	**Aciklanabilir yapay zeka**	**Tatil**	**Topolojik yapay öğrenme**	**Büyük dil modelleri**
	Çağatay Yıldız	**Özgür Martin**	**İlker Birbil**		**Çağatay Yıldız / Tolga BIrdal**	**Sergül Aydöre**
08:30-12:00	Yapay öğrenmenin temelleri Derin öğrenmenin temelleri Hayat boyu ogrenme Felaket unutma (**calisma**) Ayrisik ogrenme (**calisma**)	Derin öğrenme için optimizasyon. Rassal bayır inişi yöntemi (RBİ) RBI yakınsaklık analizi.	Açıklanabilir yapay zeka. Son işleme yaklaşımları. Optimizasyon yöntemleri		(Cagatay, 8:30-10:00) Modüler öğrenme icin grup teori (**calisma**) (Tolga, 10:00-12:00) Geometrik yapay öğrenmenin temelleri	Büyük dil modellerinin temelleri. Büyük dil modellerindeki son bilimsel gelişmeler.
12:00 - 15:00	**OGLE YEMEGI VE SERBEST ZAMAN**					
	Derin Ogrenme icin Grup teori	**Optimizasyon - guncel yontemler**		**Tatil**		**Büyük dil modelleri - ileri**
	Gönenç Onay	**Özgür Martin**	**İlker Birbil**		**Tolga Birdal**	**Sergül Aydöre**
15:00-18:00 (16:15-16:45 kek arasi)	Grup yapilari ve etkileri Derin ogrenmede kullanilian grup ornekleri Etki koruyan mimariler grup konvolusiyon teoremi	RBİ varyantları: Momentum, AdaGrad, Adam. RBİ için çizge arama ve model kurma yöntemleri.	Kuramsal çalışmalar Uygulamalar		Topolojik yapay ogrenmenin temelleri Topolojik yapay ogrenme	Büyük dil modellerindeki son bilimsel gelişmeler. Sorumlu yapay zeka açısından büyük dil modellerinin zayıflığı.
18:00 -	**AKSAM YEMEGI VE SERBEST ZAMAN**					

Hayat boyu öğrenme

1. The definition indicates five key characteristics of LL:
   (a) continuous learning process,
   (b) knowledge accumulation and maintenance in the KB,
   (c) the ability to use the accumulated past knowledge to help future learning,
   (d) the ability to discover new tasks, and
   (e) the ability to learn while working or to learn on the job.

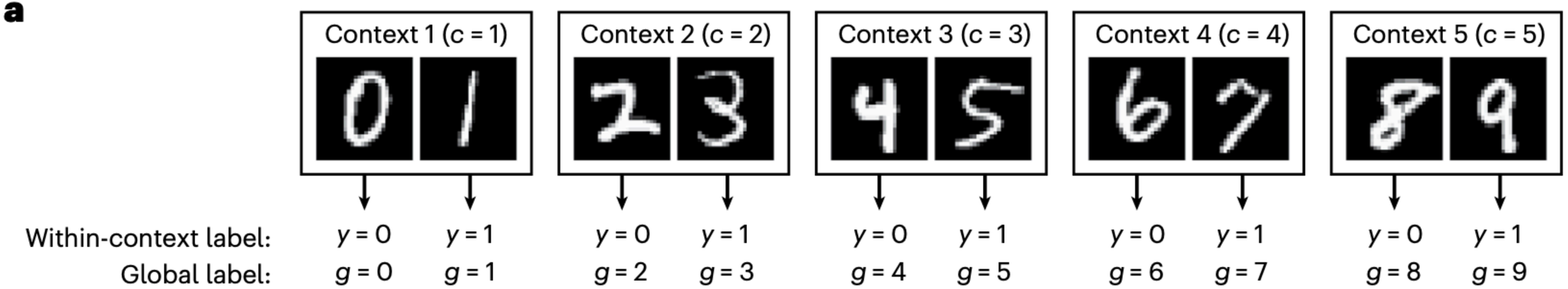
- LML sec 1.4

Hayat boyu vs diger öğrenme sekilleri

- LML sec 2

Hayat boyu öğrenme

a



b

	Input (at test time)	Expected output	Intuitive description
Task-incremental learning	Image + context label	Within-context label[a]	Choice between two digits of same context (e.g. 0 or 1)
Domain-incremental learning	Image	Within-context label	Is the digit odd or even?
Class-incremental learning	Image	Global label	Choice between all ten digits

EWC

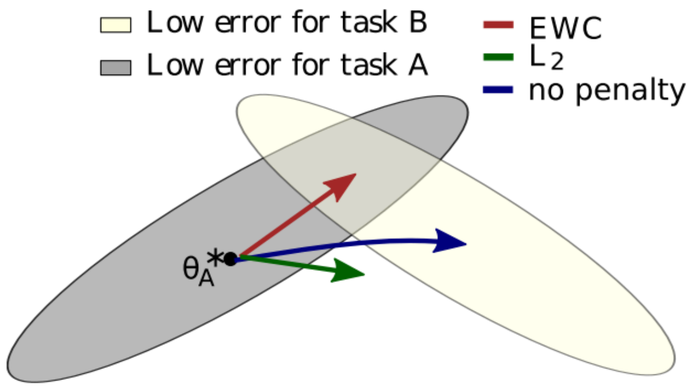


Figure 1: elastic weight consolidation (EWC) ensures task A is remembered whilst training on task B. Training trajectories are illustrated in a schematic parameter space, with parameter regions leading to good performance on task A (gray) and on task B (cream). After learning the first task, the parameters are at θ^*_A. If we take gradient steps according to task B alone (blue arrow), we will minimize the loss of task B but destroy what we have learnt for task A. On the other hand, if we constrain each weight with the same coefficient (green arrow) the restriction imposed is too severe and we can only remember task A at the expense of not learning task B. EWC, conversely, finds a solution for task B without incurring a significant loss on task A (red arrow) by explicitly computing how important weights are for task A.

approximation, the function $\mathcal{L}$ that we minimize in EWC is:

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta^*_{A,i})^2 \quad (3)$$

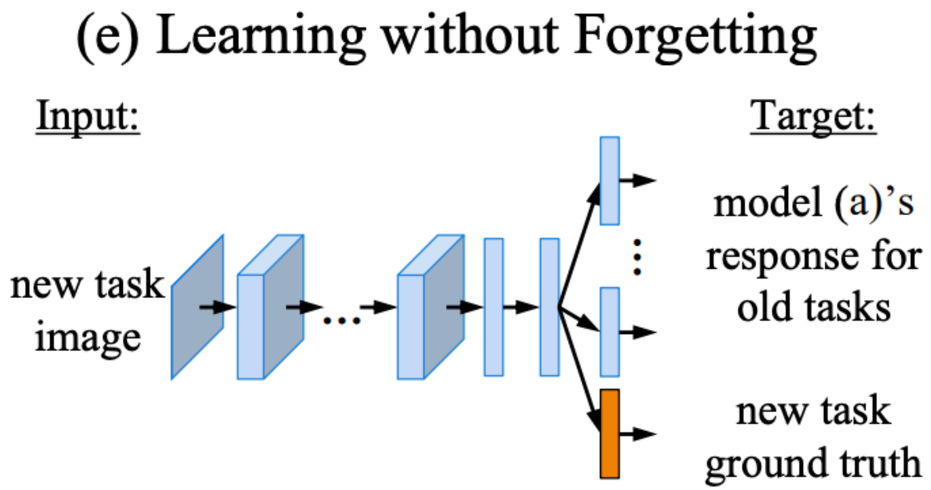
F is Fisher inf. matr.

where $\mathcal{L}_B(\theta)$ is the loss for task B only, λ sets how important the old task is compared to the new one and i labels each parameter.

In mathematical statistics, the **Fisher information** (sometimes simply called **information**[1]) is a way of measuring the amount of information that an observable random variable *X* carries about an unknown parameter *θ* of a distribution that models *X*. Formally, it is the variance of the score, or the expected value of the observed information.

LwF

- Task-specific heads
- Don’t change the output of old heads on new data



Memory replay