Regularization

We want to find an extreme of an **objective function**:

- minimizing

$$\chi^2 = \sum_k \frac{(\hat{y}_k - y_k)^2}{\sigma_k^2}$$

curve fitting

- maximizing

accuracy

$$\|Y - X\beta\|^2$$

linear regression

$$S = -\sum_i p_i \ln p_i$$

classification

$$KL(p||q) = -\int p(x) \log\left[\frac{\boldsymbol{q(x)}}{\boldsymbol{p(x)}}\right] dx$$

generation/ encoding

Regularization

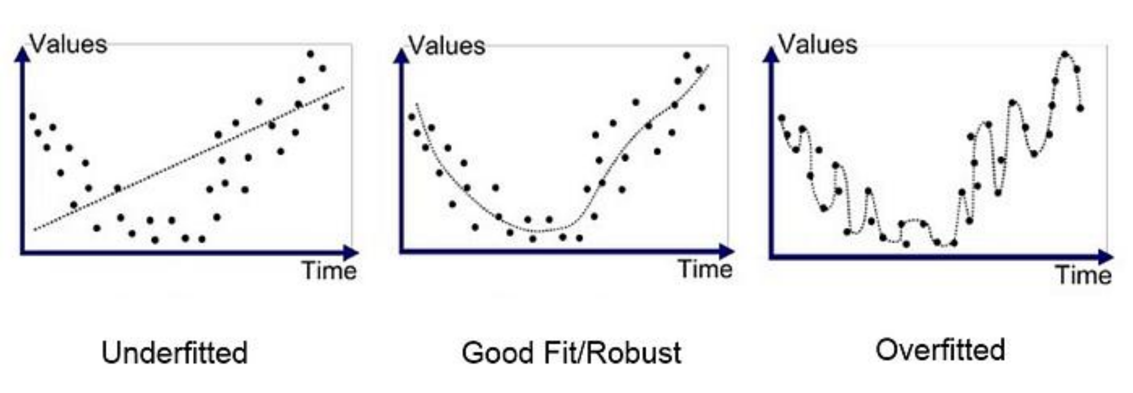
We want to find an extreme of an **objective function**:

problem: - algorithm might find an extreme, but for unreasonable values

- volume, mass, temperature (K) etc can only be positive

- values are extreme

- overfitting



credit: medium.com

- conservation:

$$S = -\sum_i p_i \ln p_i \qquad 1 = \sum_i p_i$$

Regularization

λ *Lagrangian Multiplier*

We want to find an extreme of an **objective function**:

$$\hat{\beta} = \underset{\beta}{argmin}\{\|Y - X\beta\|^2\}$$

$$\hat{\beta} = \underset{\beta}{argmin}\{\|Y - X\beta\|^2 + \boldsymbol{\lambda}\|\boldsymbol{\beta}\|^{\mathbf{1}}\}$$

the Loss Function
$\boldsymbol{L(X, Y, \lambda)}$

L1 or **L**east **a**bsolute **s**hrinkage and **s**election **o**perator
- encourages **sparsity** of β
- reduces **overfitting**

$$\hat{\beta} = \underset{\beta}{argmin}\{\|Y - X\beta\|^2 + \boldsymbol{\lambda}\|\boldsymbol{\beta}\|^{\mathbf{2}}\}$$

L2 or **Ridge**
- **penalizes large** $\boldsymbol{\beta}$

$$\hat{\beta} = \underset{\beta}{argmin}\{\|Y - X\beta\|^2 + \boldsymbol{\lambda\, max(0, -\beta)}\}$$

- **penalizes negative** $\boldsymbol{\beta}$

...and so on

Regularization

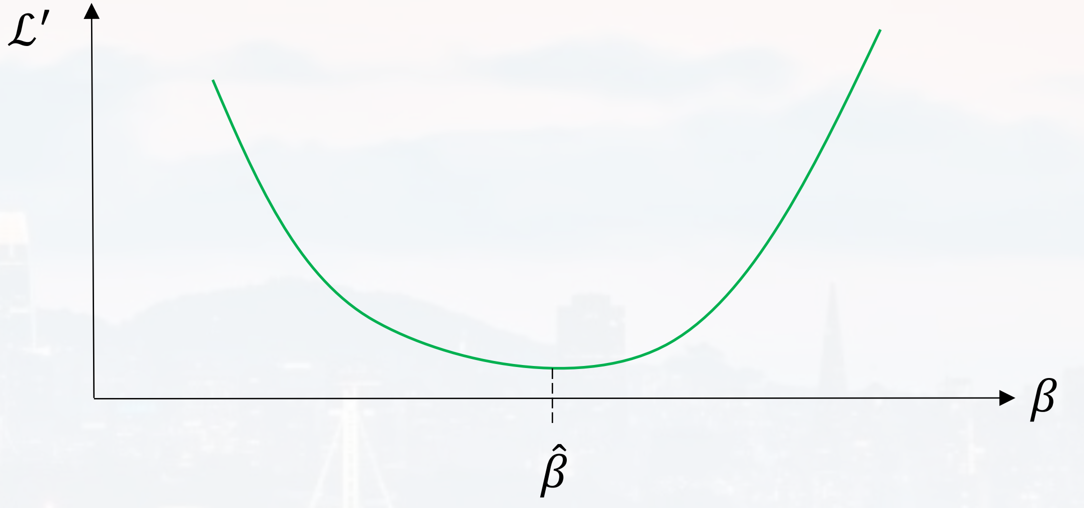
λ *Lagrangian Multiplier*

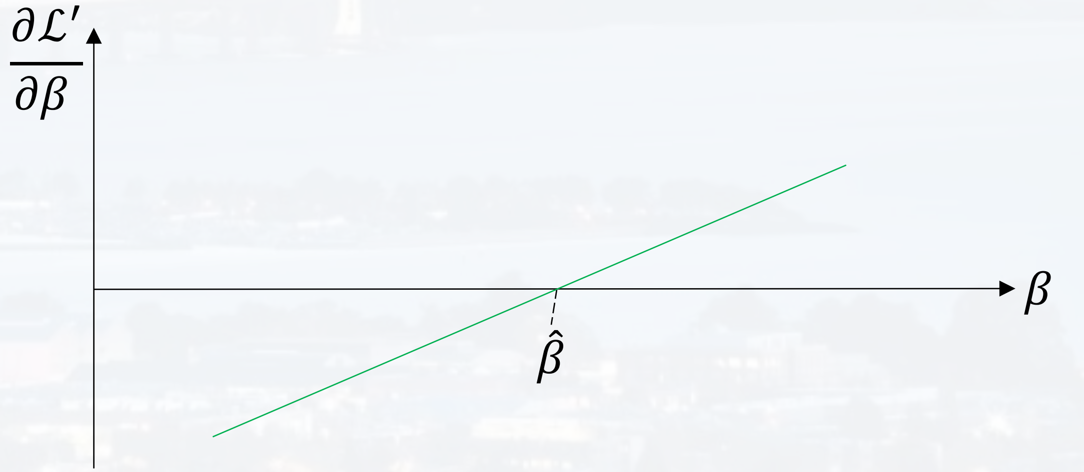
We want to find an extreme of an **objective function**:

$$\hat{\beta} = \underset{\beta}{argmin}\left\{\|Y - X\beta\|^2 + \boldsymbol{\lambda_1\|\beta\|^1} + \boldsymbol{\lambda_2\|\beta\|^2}\right\}$$

the Loss Function
$\boldsymbol{L(X, Y, \lambda)}$

$$\mathcal{L}' = \|Y - X\beta\|^2$$





Regularization

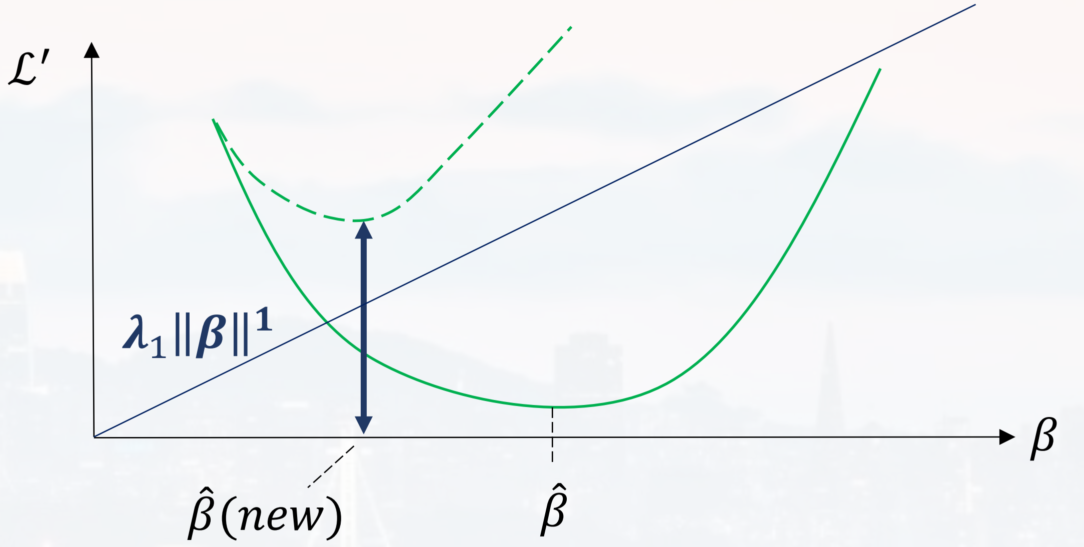
λ *Lagrangian Multiplier*

We want to find an extreme of an **objective function**:

$$\hat{\beta} = \underset{\beta}{argmin}\{\|Y - X\beta\|^2 + \boldsymbol{\lambda_1}\|\boldsymbol{\beta}\|^{\mathbf{1}} + \boldsymbol{\lambda_2}\|\boldsymbol{\beta}\|^{\mathbf{2}}\}$$

the Loss Function
$\boldsymbol{L}(\boldsymbol{X}, \boldsymbol{Y}, \lambda)$

$$\mathcal{L}' = \|Y - X\beta\|^2$$

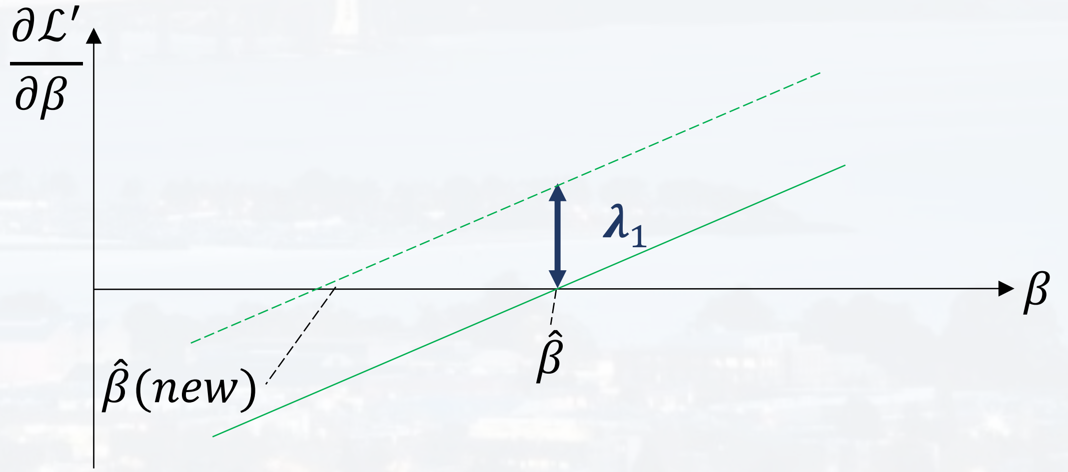


$$\frac{\partial}{\partial\beta}\left[\|Y - X\beta\|^2 + \boldsymbol{\lambda_1}\|\boldsymbol{\beta}\|^{\mathbf{1}}\right] = \frac{\partial\mathcal{L}'}{\partial\beta} + \boldsymbol{\lambda_1} sign(\boldsymbol{\beta})$$

$$= \frac{\partial\mathcal{L}'}{\partial\beta} \mp \boldsymbol{\lambda_1}$$

shifts ***if*** there is a $\boldsymbol{\beta}$, but not sensitive to its magnitude

→ large $\boldsymbol{\lambda_1}$ encourages sparsity & prevents over fitting!



Regularization

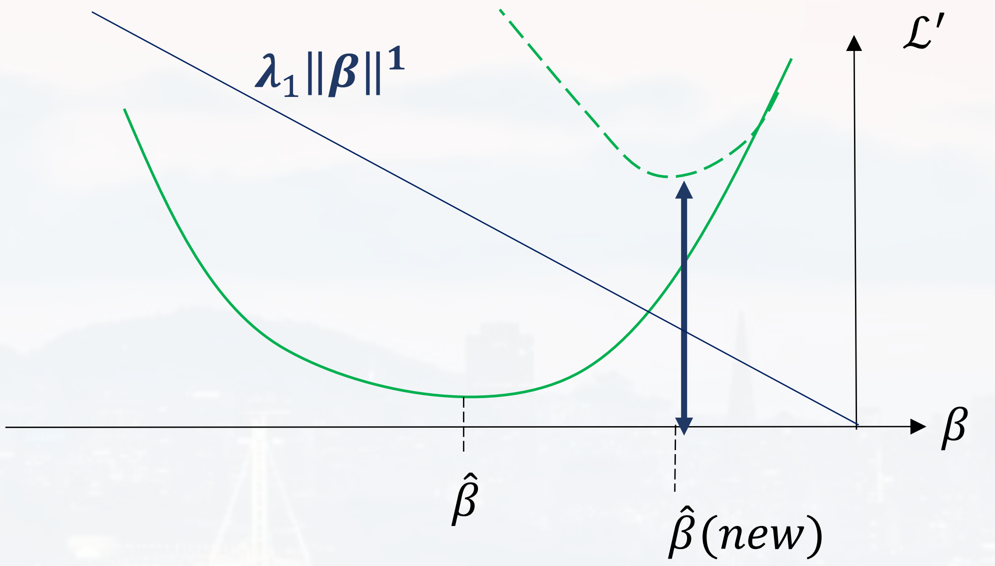
λ *Lagrangian Multiplier*

We want to find an extreme of an **objective function**:

$$\hat{\beta} = \underset{\beta}{argmin}\{\|Y - X\beta\|^2 + \boldsymbol{\lambda_1}\|\boldsymbol{\beta}\|^{\mathbf{1}} + \boldsymbol{\lambda_2}\|\boldsymbol{\beta}\|^{\mathbf{2}}\}$$

the Loss Function
$\boldsymbol{L(X, Y, \lambda)}$

$$\mathcal{L}' = \|Y - X\beta\|^2$$

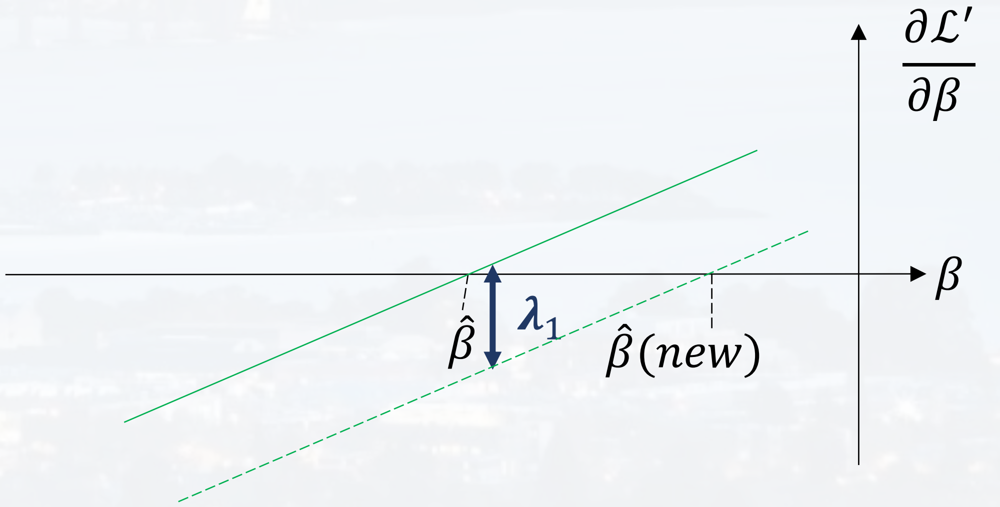


$$\frac{\partial}{\partial\beta}\left[\|Y - X\beta\|^2 + \boldsymbol{\lambda_1}\|\boldsymbol{\beta}\|^{\mathbf{1}}\right] = \frac{\partial\mathcal{L}'}{\partial\beta} + \boldsymbol{\lambda_1} sign(\boldsymbol{\beta})$$

$$= \frac{\partial\mathcal{L}'}{\partial\beta} \mp \boldsymbol{\lambda_1}$$

shifts ***if*** there is a $\boldsymbol{\beta}$, but not sensitive to its magnitude

→ large $\boldsymbol{\lambda_1}$ encourages sparsity & prevents over fitting!



Regularization

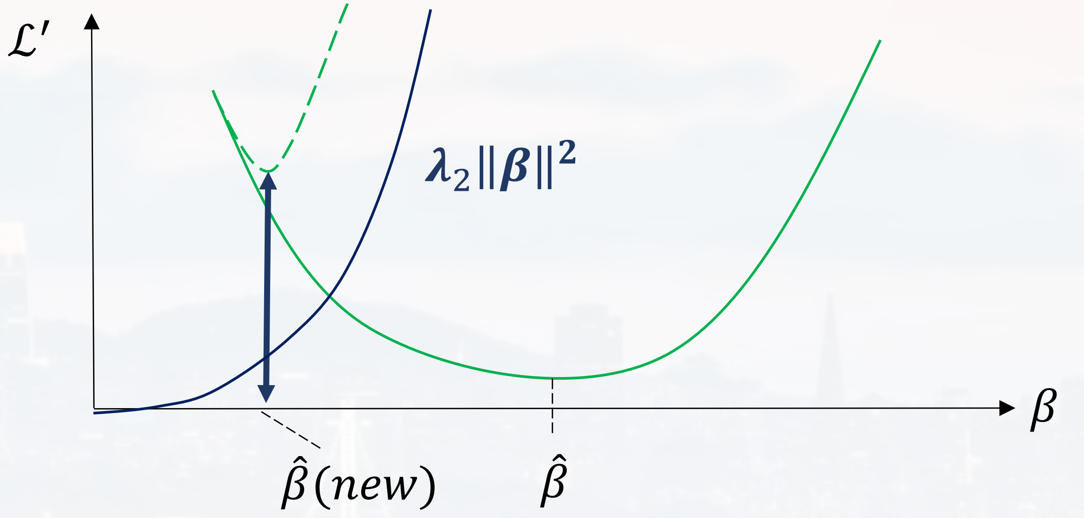
λ *Lagrangian Multiplier*

We want to find an extreme of an **objective function**:

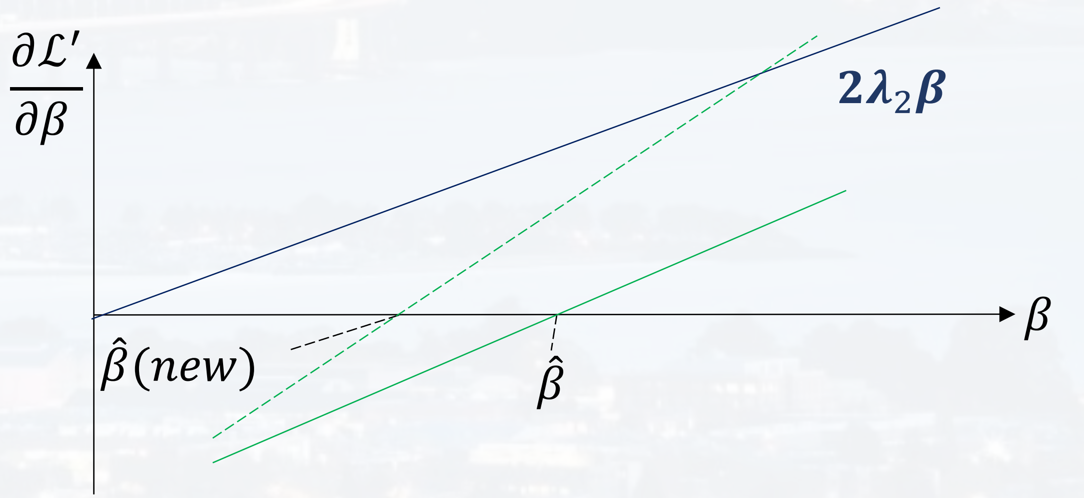
$$\hat{\beta} = \underset{\beta}{argmin} \{\|Y - X\beta\|^2 + \boldsymbol{\lambda_1\|\beta\|^1} + \boldsymbol{\lambda_2\|\beta\|^2}\}$$

the Loss Function
$\boldsymbol{L(X, Y, \lambda)}$

$$\mathcal{L}' = \|Y - X\beta\|^2$$



$$\frac{\partial}{\partial \beta}\left[\|Y - X\beta\|^2 + \boldsymbol{\lambda_2\|\beta\|^2}\right] = \frac{\partial \mathcal{L}'}{\partial \beta} + \boldsymbol{2\lambda_2\beta}$$



shifts according to the **magnitude** of $\boldsymbol{\beta}$,

→ large $\boldsymbol{\lambda_2}$ encourages smaller magnitudes for $\boldsymbol{\beta}$

Regularization

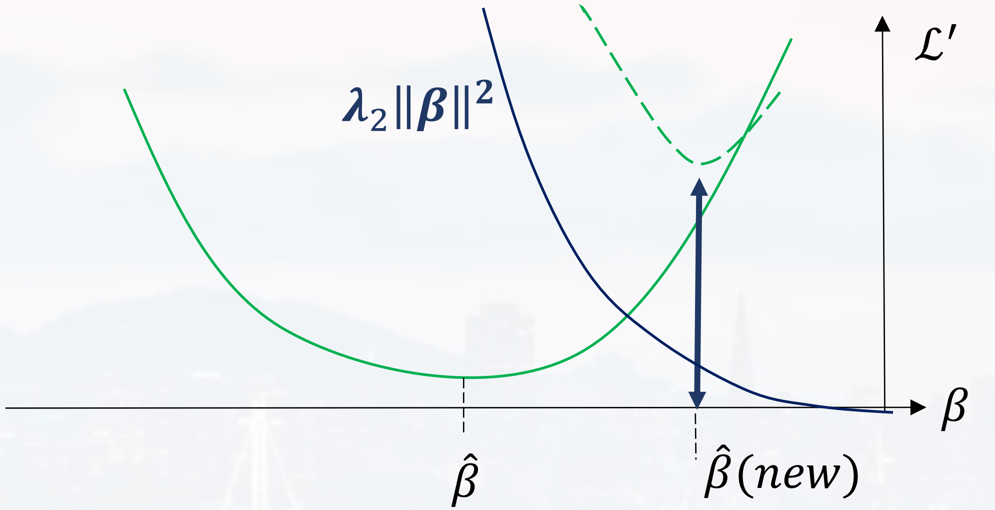
We want to find an extreme of an **objective function**:

λ *Lagrangian Multiplier*

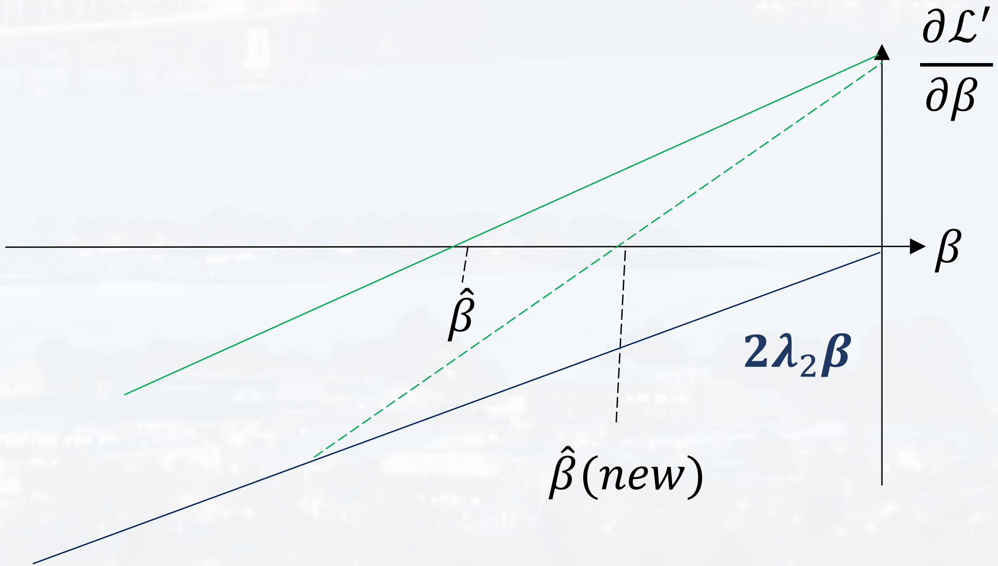
$$\hat{\beta} = \underset{\beta}{argmin}\{\|Y - X\beta\|^2 + \boldsymbol{\lambda_1\|\beta\|^1} + \boldsymbol{\lambda_2\|\beta\|^2}\}$$

the Loss Function
$\boldsymbol{L(X, Y, \lambda)}$

$$\mathcal{L}' = \|Y - X\beta\|^2$$



$$\frac{\partial}{\partial\beta}[\|Y - X\beta\|^2 + \boldsymbol{\lambda_2\|\beta\|^2}] = \frac{\partial\mathcal{L}'}{\partial\beta} + \boldsymbol{2\lambda_2\beta}$$



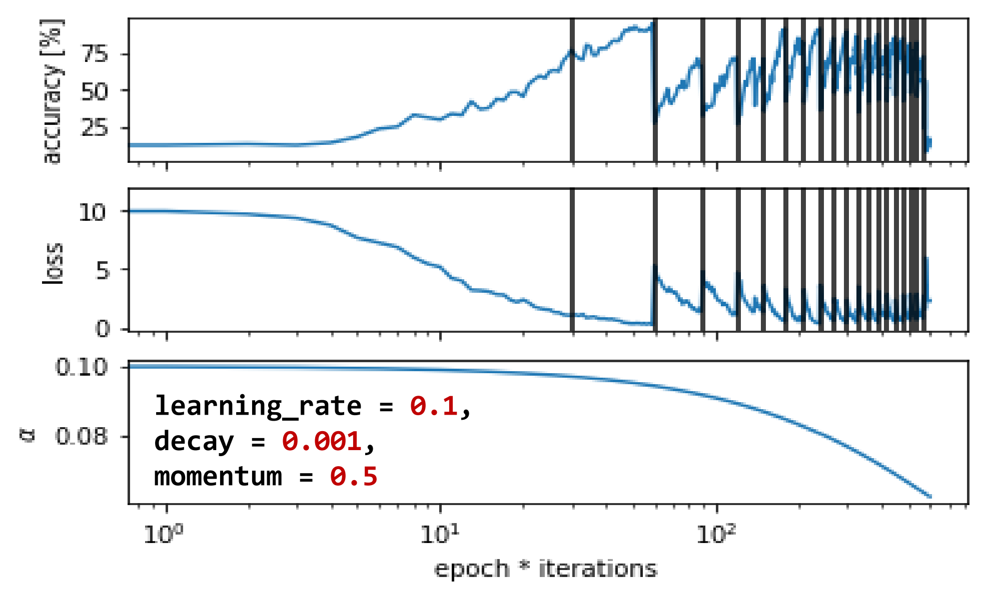
shifts according to the **magnitude** of $\boldsymbol{\beta}$,

→ large $\boldsymbol{\lambda_2}$ encourages smaller magnitudes for $\boldsymbol{\beta}$

Regularization

note:

- how to implement L1 and L2 regularization for lin. regression in Python depends on the library (see documentation)
- "Elastic Net" balances L1 and L2: $\lambda\left(\frac{1-\alpha}{2}\|\boldsymbol{\beta}\|^2+\alpha\|\boldsymbol{\beta}\|^1\right)$
- L1: deals with highly correlated factors (sparicty)
- L2: deals with large facors (keeps solution stable)
- regularisation is pretty common, not only for lin. models



LeNet numpy only

```
W = np.load('weightsC1.npy')
W[:,:,0]
```

	0	1	2	3	4	5
0	-15923.3	-16647.4	-16277.1	-16993.1	-15715.7	-15390.1
1	9795.1	8468.91	9110.83	8205.6	9852.06	11061
2	37956.4	36572.6	37324.2	37008.4	37485.4	38515.2
3	39686.6	39131	39087.4	39614.6	39421.7	39876.7
4	25465.3	26270.1	25232.4	25836.7	25590.5	25270.9

Regularization

note: the

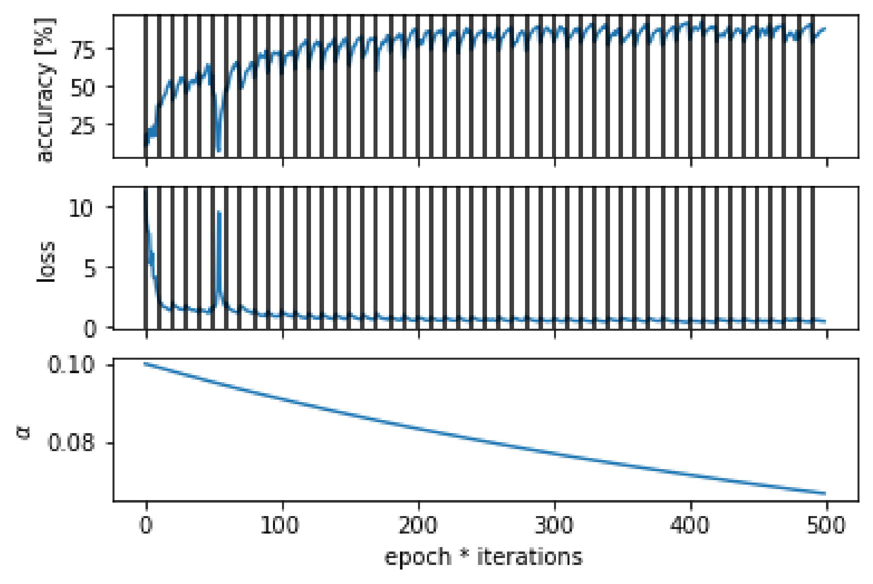
- how to implement L1 and L2 regularization for lin. regression in Python depends on library (see documentation)

- "Elastic Net" balances L1 and L2: $\lambda\left(\frac{1-\alpha}{2}\|\boldsymbol{\beta}\|^{2} + \alpha\|\boldsymbol{\beta}\|^{1}\right)$

- L1: deals with highly correlated factors (sparicty)

- L2: deals with large facors (keeps solution stable)

- regularisation is pretty common, not only for lin. models

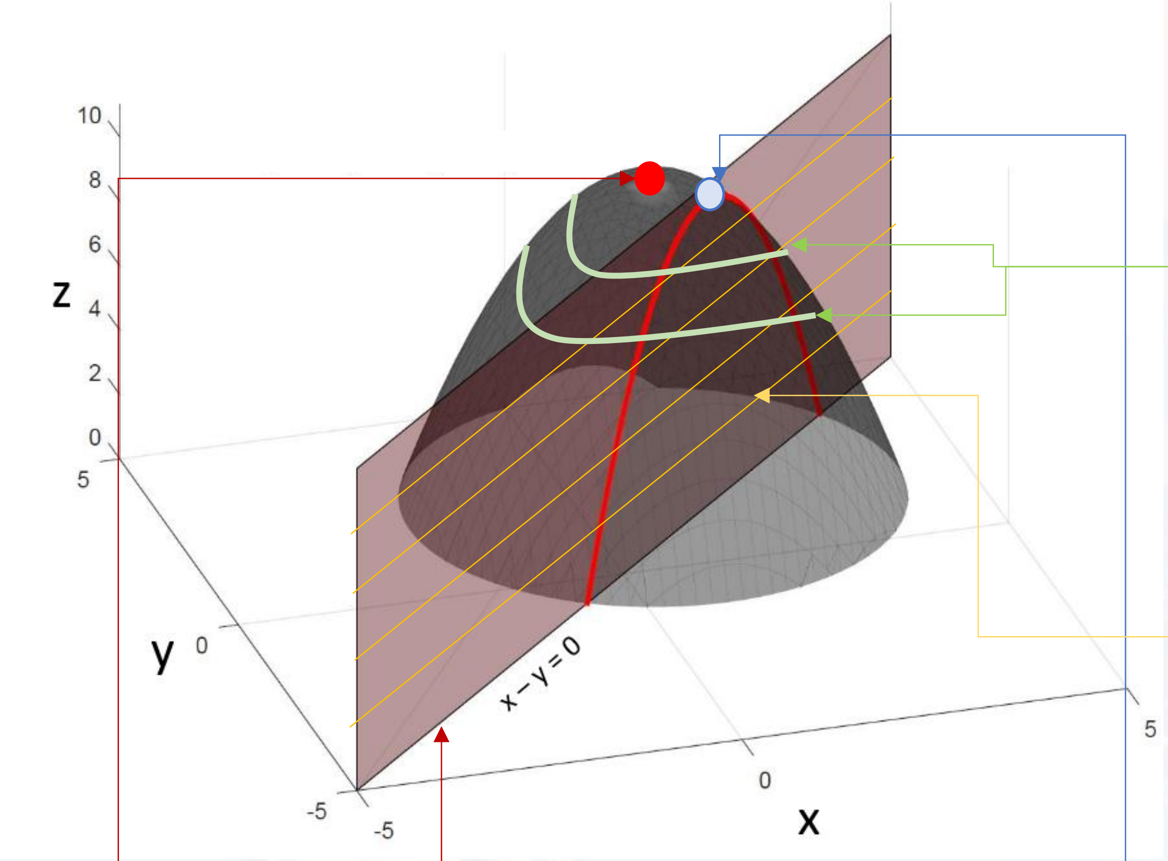


LeNet numpy only

L1 and L2 regualrization

Regularization

More about ***Lagrangian Multiplier***:



$z = 10$ at
$x = 2$
$y = 1$

constrain $g(x,y) = x - y = 0$

maximum of the function

$$f(x,y) = z = -(x-2)^2 - (y-1)^2 + 10$$

level lines $f(x,y) = const$

$$df(x,y) = \frac{\partial f(x,y)}{\partial x}dx + \frac{\partial f(x,y)}{\partial y}dy = 0$$

$$= gradf \; d\vec{r} = 0$$

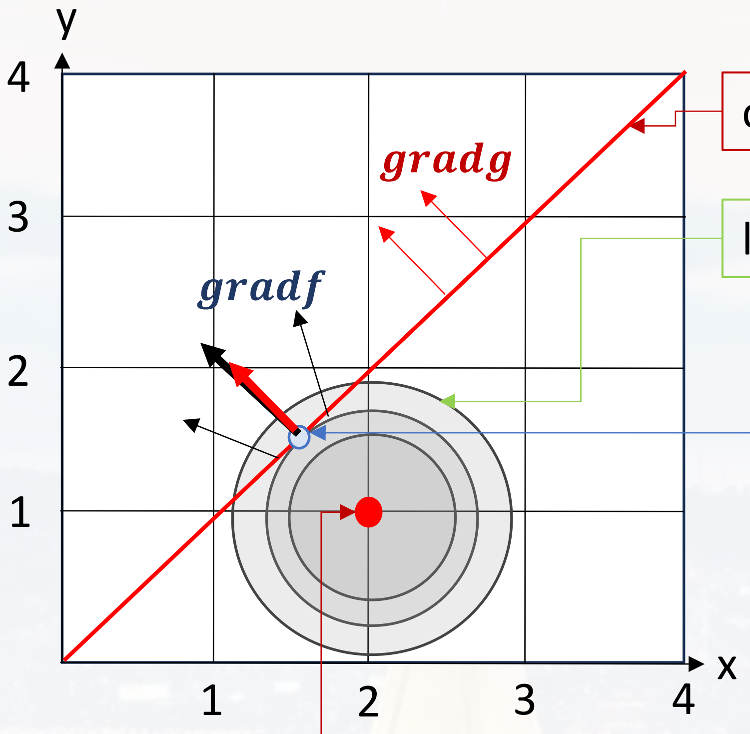
level lines $g(x,y) = const$

$$dg(x,y) = \frac{\partial g(x,y)}{\partial x}dx + \frac{\partial g(x,y)}{\partial y}dy = 0$$

$$= gradg \; d\vec{r} = 0$$

maximum of the function, subject to $g(x,y)$

Regularization



constrain $g(x,y) = x - y = 0$

level lines $f(x,y) = const$

$z = 10$ at
$x = 2$
$y = 1$

maximum of the function, subject to $g(x,y)$

the maximum of $f(x,y)$ subject to $g(x,y)$ located where:

$$df(x,y) \quad = dg(x,y)$$

$$gradf \; d\vec{r} = gradg \; d\vec{r}$$

$$gradf \quad = gradg$$

Both gradients need to point in the same direction
(hence, can be multiplied with a constant, say λ)!

$$\boxed{gradf \quad = \lambda \; gradg}$$

λ **Lagrangian Multiplier**

Regularization

the maximum of $f(x,y)$ subject to $g(x,y)$

$$grad f = \lambda \, grad g$$

$$f(x,y) - \lambda g(x,y) = const$$

the Lagrangian
$\boldsymbol{L(x,y,\lambda)}$

more general:

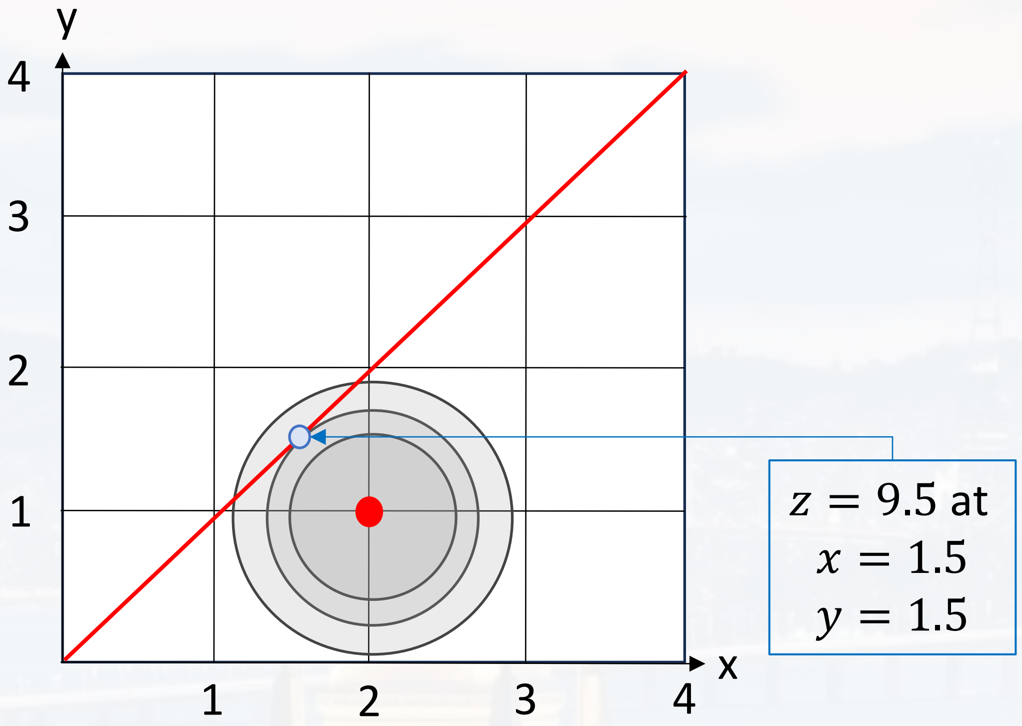
$$\boxed{L(x_1, x_2, \dots, x_i, x_N, \lambda_1, \lambda_2, \dots, \lambda_k, \lambda_K) = f(x_1, x_2, \dots, x_i, x_N) - \sum_{k=1}^{K} \lambda_k \, g_k(x_1, x_2, \dots, x_i, x_N)}$$

note: - ***N* dimensions and $K \leq N$ constrains**

- we need to solve *N* (from the gradient) + *K* equations by using the constrains
- optimization: more robust results (most common L1 and L2 regularization, as before)
- machine learning: including constrains in loss function (see later)

Regularization

the maximum of $f(x,y)$ subject to $g(x,y)$



maximum of the function

$$f(x,y) = z = -(x-2)^2 - (y-1)^2 + 10$$

constrain $g(x,y) = x - y = 0$

constrain $x = y$

$x = 1.5$
$y = 1.5$

$f(1.5, 1.5) = 9.5$

$$grad f = \lambda\, grad g$$

$$\frac{\partial f(x,y)}{\partial x} = \lambda \frac{\partial g(x,y)}{\partial x} \qquad -2(x-2) = \lambda$$

$$\frac{\partial f(x,y)}{\partial y} = \lambda \frac{\partial g(x,y)}{\partial y} \qquad -2(y-1) = -\lambda$$

$$y = -x + 3$$

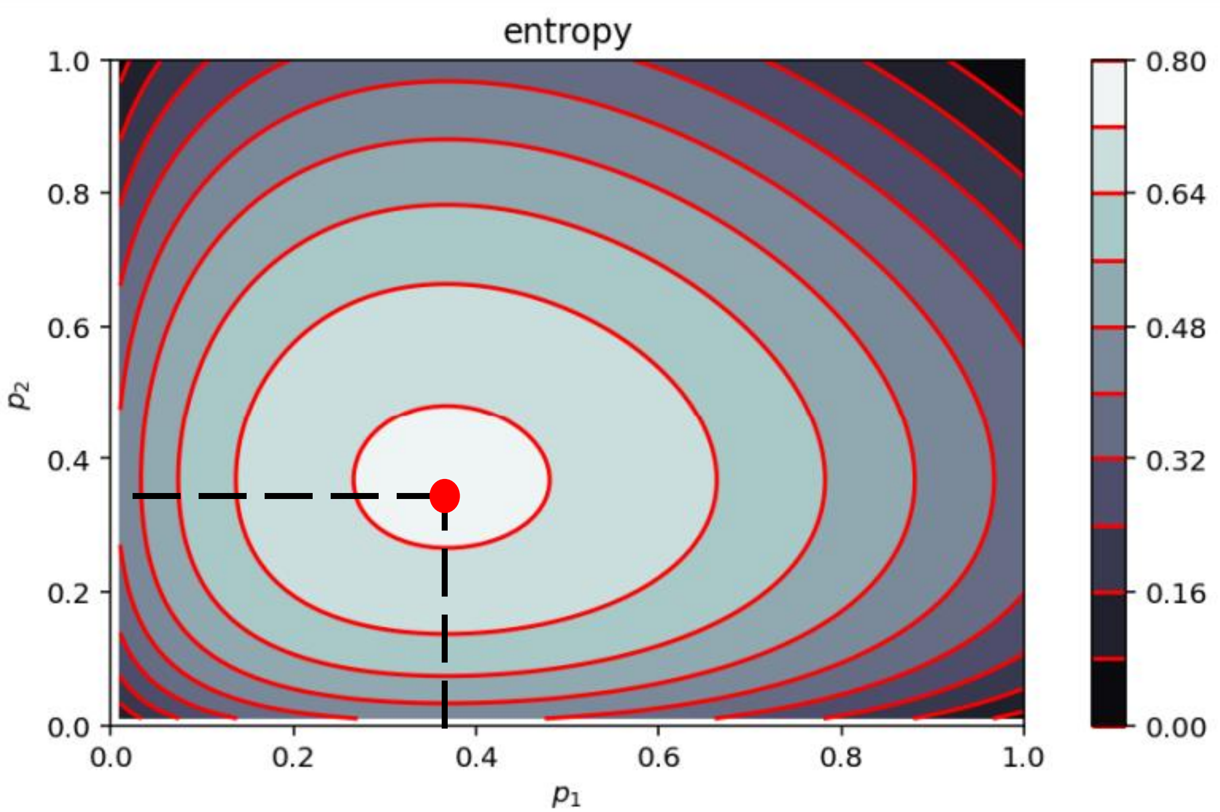
Regularization

maximum entropy of flipping a coin:

$$f(p_1, p_2) = -p_1 \ln p_1 - p_2 \ln p_2$$

subject to

$$g(p_1, p_2) = p_1 + p_2 = 1$$



absolute maximum:

$$\frac{\partial f(p_1, p_2)}{\partial p_1} = 0 \qquad \frac{\partial f(p_1, p_2)}{\partial p_2} = 0$$

$$-\ln p_1 - 1 = 0 \qquad -\ln p_2 - 1 = 0$$

$$p_1 = p_2 = \frac{1}{e}$$

$$f\left(\frac{1}{e}, \frac{1}{e}\right) = \frac{2}{e} \approx 0.74$$

$$p_1 + p_2 = \frac{2}{e} > 1$$

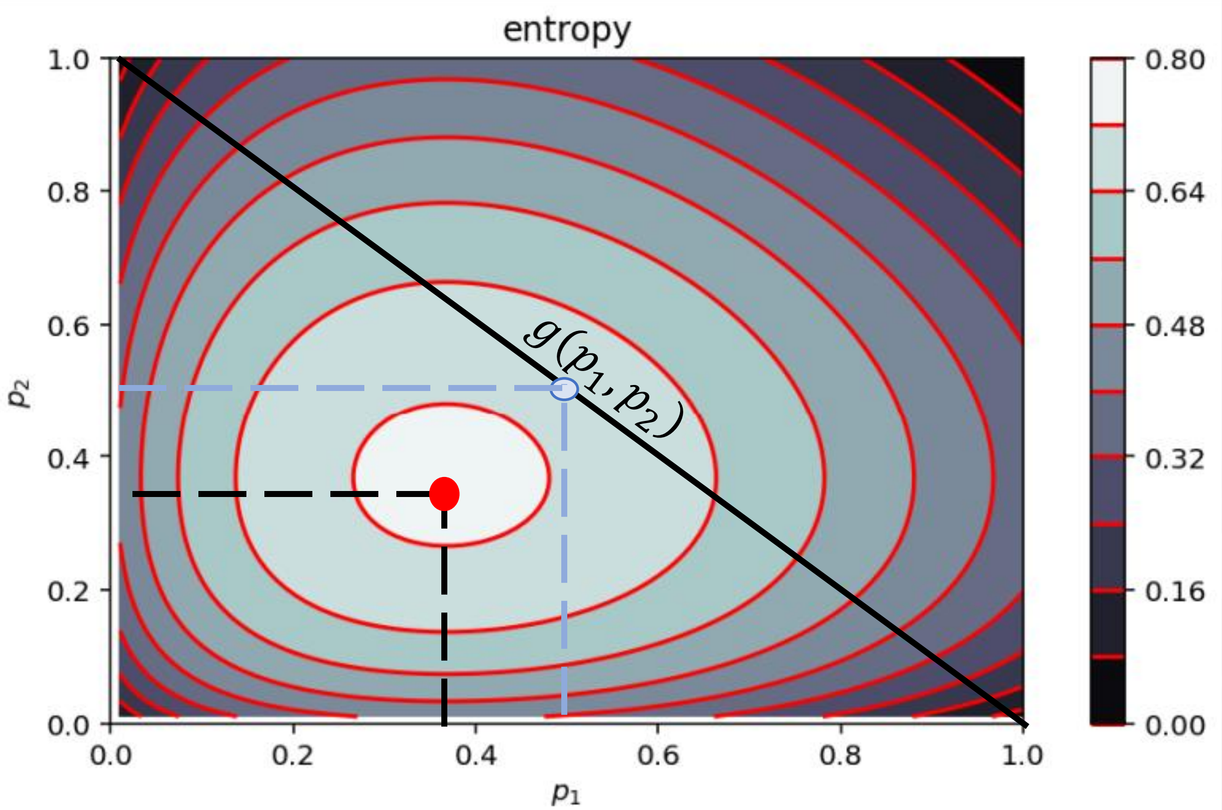
Regularization

maximum entropy of flipping a coin:

$$f(p_1, p_2) = -p_1 \ln p_1 - p_2 \ln p_2$$

subject to

$$g(p_1, p_2) = p_1 + p_2 = 1$$



maximum subject to $g(p_1, p_2)$:

$$\frac{\partial f(p_1, p_2)}{\partial p_1} = \lambda \frac{\partial g(p_1, p_2)}{\partial p_1}$$

$$\frac{\partial f(p_1, p_2)}{\partial p_2} = \lambda \frac{\partial g(p_1, p_2)}{\partial p_2}$$

$$-\ln p_1 - 1 = \boldsymbol{\lambda} \qquad -\ln p_2 - 1 = \boldsymbol{\lambda}$$

$$\boldsymbol{p_1 = p_2}$$

constrain: $p_1 + p_2 = 1$

$$p_1 = p_2 = \frac{1}{2}$$

$$f\left(\frac{1}{2}, \frac{1}{2}\right) = ln2 \approx 0.69$$

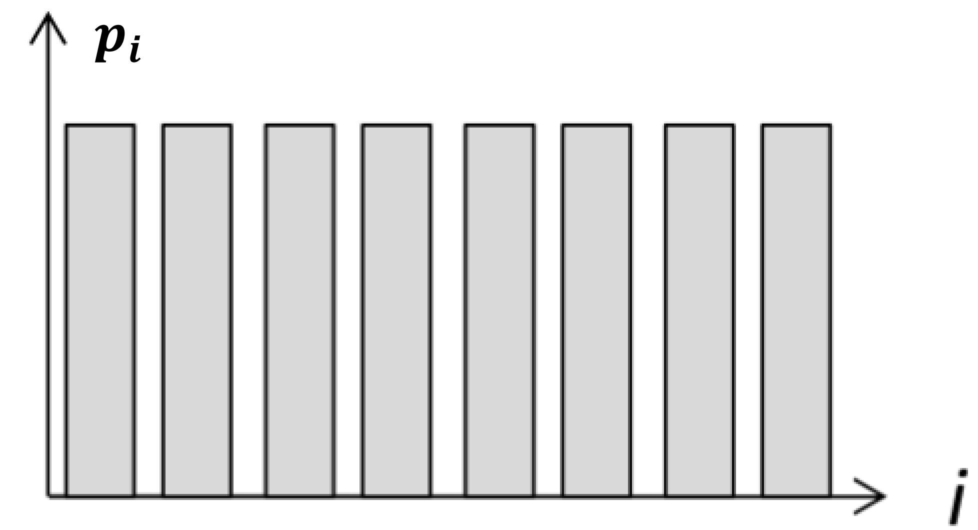
Regularization

maximum entropy for I states:

$$f(p_1,..,p_i,...p_I) = -\sum_{i=1}^{I} p_i \, ln p_i$$

subject to

$$g(p_1,..,p_i,...p_I) = \sum_{i=1}^{I} p_i \; = 1$$



maximum subject to $g(p_1,..,p_i,...p_I)$:

$$\frac{\partial f(p_1,..,p_i,...p_I)}{\partial p_i} = \lambda \frac{\partial g(p_1,..,p_i,...p_I)}{\partial p_i}$$

$$-\ln p_i - 1 = \lambda \qquad p_i = e^{-(1+\lambda)}$$

constrain: $$\sum_{i=1}^{I} e^{-(1+\lambda)} = 1$$

$$I e^{-(1+\lambda)} = 1$$

$$e^{-(1+\lambda)} = \frac{1}{I}$$

probabilities are constant!
→ flat distribution!

$$\boxed{p_i = \frac{1}{I}}$$

Regularization

N:	number of indistinguishable particles
n_i:	number of particles in micro state i
I:	number of states
p_i:	probability of a particle being in micro state I
ε_i:	energy in state i

maximum entropy for I states:

$$f(p_1, .., p_i, \dots p_I) = -\sum_{i=1}^{I} p_i \, ln p_i$$

subject to

$$g_1(p_1, .., p_i, \dots p_I) = \sum_{i=1}^{I} p_i \;= 1$$

if *N* and ***total energy*** is conserved

$$g_2(p_1, .., p_i, \dots p_I) = \sum_{i=1}^{I} p_i \varepsilon_\mathrm{i} = \frac{E_{tot}}{N} = \frac{1}{N}\sum_{i=1}^{I} n_i \varepsilon_\mathrm{i}$$

$$\frac{\partial f(p_1, .., p_i, \dots p_I)}{\partial p_i} = \lambda_1 \frac{\partial g_1(p_1, .., p_i, \dots p_I)}{\partial p_i} + \lambda_2 \frac{\partial g_2(p_1, .., p_i, \dots p_I)}{\partial p_i}$$

$$-\ln p_i - 1 = \lambda_1 + \lambda_2 \frac{\partial \sum_{j=1}^{I} p_j \varepsilon_\mathrm{j}}{\partial p_i}$$

Regularization

Lagrangian Multiplier
Examples

N:	**number of indistinguishable particles**
n_i:	**number of particles in micro state *i***
I:	**number of states**
p_i:	**probability of a particle being in micro state *I***
ε_i:	**energy in state *i***

maximum entropy for I states:

$$f(p_1, .., p_i, \dots p_I) = -\sum_{i=1}^{I} p_i \, ln p_i$$

subject to

$$g_1(p_1, .., p_i, \dots p_I) = \sum_{i=1}^{I} p_i \; = 1$$

if *N* and ***total energy*** is conserved

$$g_2(p_1, .., p_i, \dots p_I) = \sum_{i=1}^{I} p_i \varepsilon_i$$

$$-\ln p_i - 1 = \lambda_1 + \lambda_2 \frac{\partial \sum_{j=1}^{I} p_j \varepsilon_j}{\partial p_i}$$

$$-\ln p_i - 1 = \lambda_1 + \lambda_2 \varepsilon_i$$

$$p_i = e^{-(1+\lambda_1)} \, e^{-\lambda_2 \varepsilon_i}$$

from g_1:

$$p_i = \frac{1}{\sum_{i=1}^{I} e^{-\lambda_2 \varepsilon_i}} \, e^{-\lambda_2 \varepsilon_i}$$

partition function Z

$$Z = \sum_{i=1}^{I} e^{-\lambda_2 \varepsilon_i}$$

Boltzmann distribution $\quad \boxed{p_i = \frac{1}{Z} \, e^{-\lambda_2 \varepsilon_i}}$

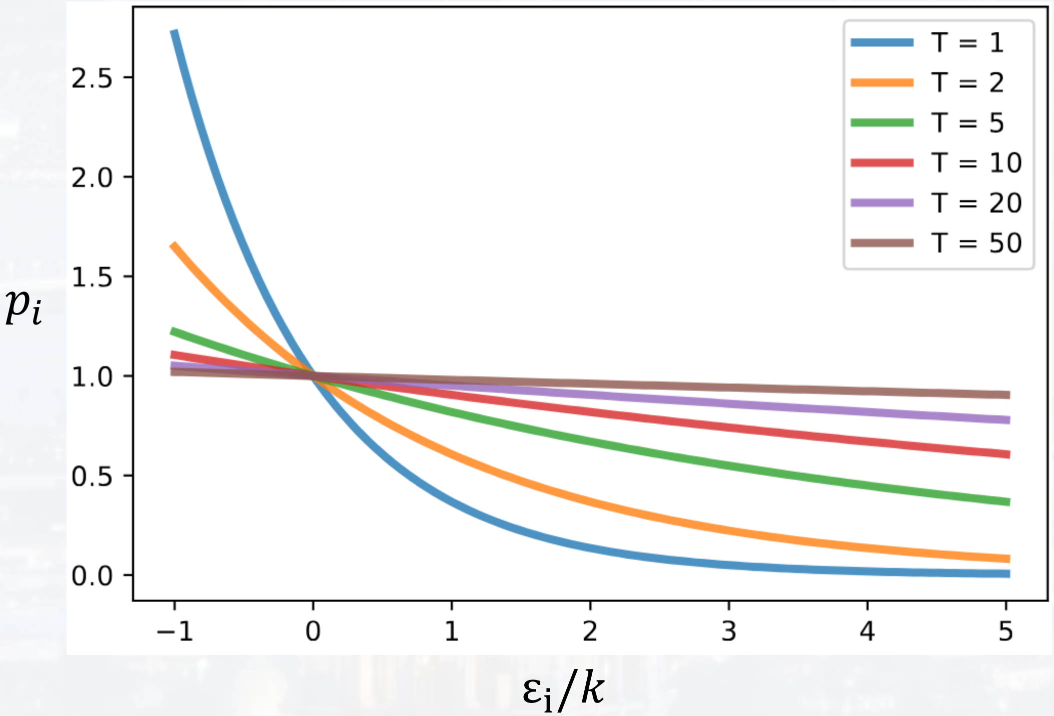
Regularization

partition function Z $$Z = \sum_{i=1}^{I} e^{-\lambda_2 \varepsilon_i}$$

Boltzmann distribution $$\boxed{p_i = \frac{1}{Z}\, e^{-\lambda_2 \varepsilon_i}}$$

one can show: $$\lambda_2 = \frac{1}{kT}$$

N:	number of indistinguishable particles
n_i:	number of particles in micro state *i*
I:	number of states
p_i:	probability of a particle being in micro state *I*
ε_i:	energy in state *i*



note:

- for $T \to \infty$, $Z \to I$, i. e. higher states become more accessible and $p_i \to \frac{1}{I}$
- we used maximum entropy: **equilibrium** state for large *N*
- *N* **and** E_{tot} are constant
- ANN: **softmax layer** for classification probabilities (see later)

Regularization