

## LR - SGD

- Musím ručně vyhledávat, kde je optimum mezi under/over fittingem.
- Pokud je vývoj test. erroru „čím“, už můžu ukončit learning.
- Například když už je náhon, je proče

Obecně „trainovat model“ = iterovat s jinými parametry

## Features - tuning:

- ↳ extrahuje-ličejších dat pro klasifikaci
- > zejména pokud máme málo dat, chce co nejlepší klasifikační výsly

- prakticky jde o jeden jediný vektorůch vektorůch vah z prezentace

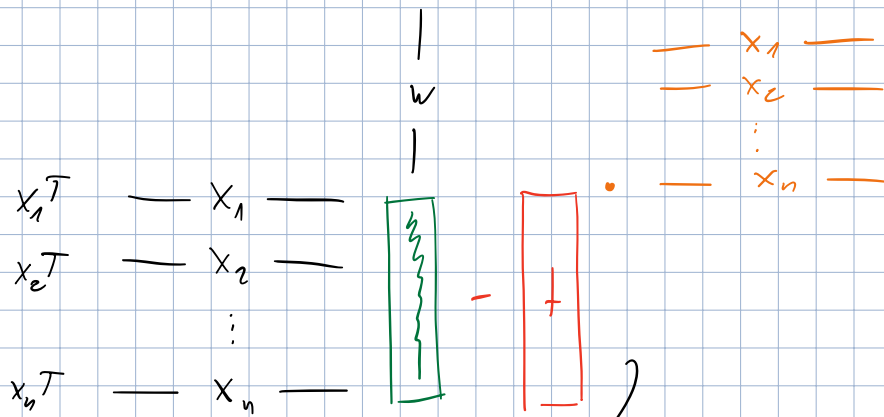
## Polynomial features:

- násobím vstupny mezi sebou, čímž si vytvořím features pro všechny možné kombinace.
- hrozí to ale zvětšuje objem (exponenciálně)
  - pozor, E2 overfitting

## One-hot feature

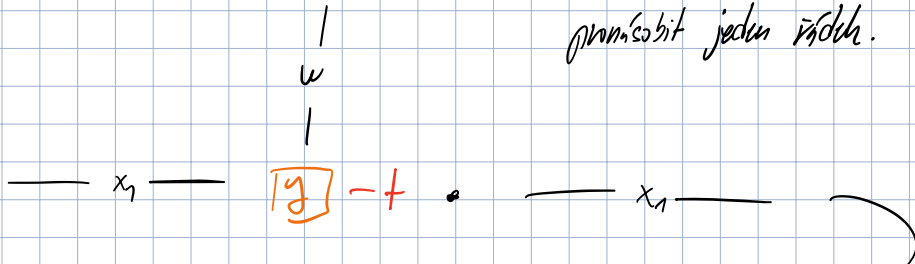
- pokud například trénuji, který den znám je, nemůžu si je 0 indexovat, protože tenhle index se násobí váhami a to nejde.
- takže si vytvořím N binárních features, kde 1 je prave u  $x_i$ , které chce reprezentovat. Takže feature reprezentující středu je ... 0,0,1,0,0,0,0
- ↳ = takže jsem to appendoval ke tomu vstupu
- pokud toho bude moc, třeba všechny dny v roce, tak je to nebezpečné pro overfitting.

$$\propto \frac{1}{B} \cdot \sum_{i \in B} \left( (x_i^T w - t_i) x_i \right)$$



$$(x_i^T w - t)$$

→ V pythonu to bude obzř \*  
protože chci jedním iteračním  
přemnožit jeden vektor.



→ na konci dostanu opět vektor

jako  $x_i$ , jen bude přemnožený upravenou váhou  
vzdělání předchozí.