ML TUTORIAL 2 – BACKPROPAGATION, OPTIMIZERS

HOW TO LEARN WEIGHTS OF AN MLP?

$$\bar{y} = \dots \sigma\left(\sigma\left(\bar{x}\,\underline{W}_1 + \bar{b}_1\right)\underline{W}_2 + \bar{b}_2\right) \dots$$

$$E = \frac{1}{2}\left(\bar{y} - \hat{\bar{y}}\right)^T\left(\bar{y} - \hat{\bar{y}}\right)$$ ← MSE LOSS, VECTORIZED

WE NEED $\frac{\partial E}{\partial \underline{W}_1}, \frac{\partial E}{\partial \underline{W}_2}, \dots, \frac{\partial E}{\partial \bar{b}_1}, \frac{\partial E}{\partial \bar{b}_2} \dots$

THESE GRADIENTS ARE THEN USED FOR GRADIENT DESCENT, TO MODIFY THE CURRENT PARAMETERS:

$$\underline{W}_1^{NEW} = \underline{W}_1^{OLD} - \eta \frac{\partial E}{\partial \underline{W}_1}$$

IDEA: USE CHAIN RULE

EXAPLE:

$$y = h(g(f(x)))$$

IMMEDIATE VARIABLES

$$u = f(x)$$
$$v = g(f(x))$$

$$\frac{dy}{dx} = \frac{dy}{dv}\frac{dv}{du}\frac{du}{dx}$$

ANOTHER NOTATION THAT CAN BE USED IN CASE IT IS CLEAR FROM THE CONTEXT WHICH VARABLE TO USE:

$$\frac{d\,h(g(f(x)))}{dx} = h'(g(f(x)))\,g'(f(x))\,f'(x)$$

WHERE $f'(x) = \frac{df(x)}{dx}$, $g'(x) = \frac{dg(x)}{dx}$, ...

FROM THE NOTATION

$$h'(\overbrace{g(f(x))}^{v})\, g'(\overbrace{f(x)}^{u})\, f'(x)$$

IT CAN BE SEEN, THAT THE PROCESS CAN BE DIVIDED IN 2 PARTS:

1. CALCULATE THE VALUE OF ALL IMMEDIATE VARIABLES AND SAVE THEM:

$$u = f(x) \qquad v = g(u) \qquad y = h(v)$$

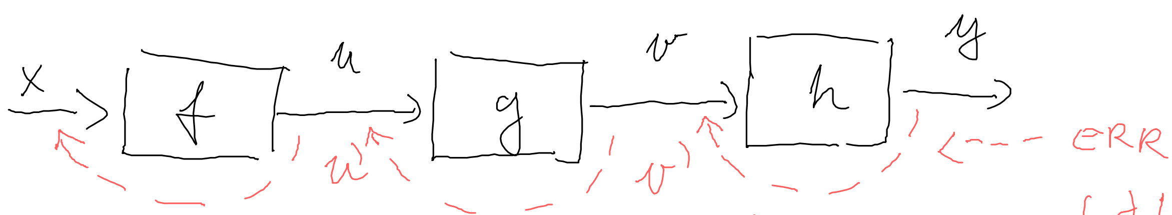
THIS IS THE <u>FORWARD PASS</u>

COMPUTATION GRAPH:

$$\xrightarrow{x} \boxed{f} \xrightarrow{u} \boxed{g} \xrightarrow{v} \boxed{h} \xrightarrow{y}$$

② CALCULATE THE DERIVATIVE OF EACH FUNCTION INDEPENDENTLY AT THE POINT OF THEIR ARGUMENT IN FORWARD PASS. THEN MULTIPLY THEM TOGETHER.

THIS IS THE BACKWARD PASS



$\left(\frac{\partial L}{\partial y}\right)$

LAGRANGE NOTATION: $f'(x)\cdot u'$ $\quad g'(u)\, v'$ $\quad h'(v)\cdot \text{ERR}$

LEIBNIZ NOTATION: $\frac{\partial u}{\partial x}\, u'$ $\quad \frac{\partial v}{\partial u}\, v'$ $\quad \frac{\partial y}{\partial v}\, \text{ERR}$

NOTE: ALL OF THEM ARE LOCAL COMPUTATION, THEY ONLY DEPEND ON THE ARGUMENT OF THE FORWARD PASS, AND THE ERROR TERM OF THE BACKWARD PASS

ARGUMENTS (ACTIVATIONS) OF FORWARD PASS $\longrightarrow$

GRADIENTS FROM THE BACKWARD PASS $\longleftarrow$

THE LOCAL COMPUTATION DOESN'T CARE HOW THE INPUT IS CREATED OR HOW THE OUTPUT GRAD IS FORMED: THEY ALWAYS HAVE THE <u>SAME FORM</u>. THIS ENABLES THE FUNCTIONS TO BE CHAINED TOGETHER LIKE BUILDING BLOCKS, INDEPENDENTLY OF EACH OTHER. PYTORCH / TENSOR-FLOW IS JUST A LIBRARY OF SUCH BLOCKS.

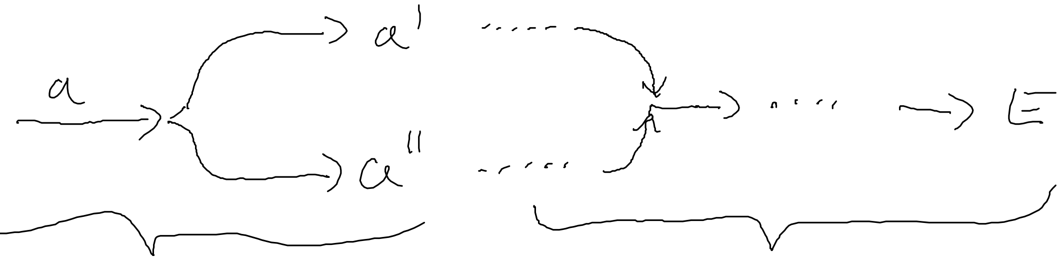
IN PRACTICE, WE USE A VECTORIZED FORM.

<u>RECIPE:</u>

1. START FROM THE BACK, GO TO THE FRONT

2. CALCULATE GRADS FOR ALL INPUTS OF THE BLOCK

3. IF THERE IS A FORK, ADD GRADS TOGETHER (MULTI-VARIATE CHAIN RULE)



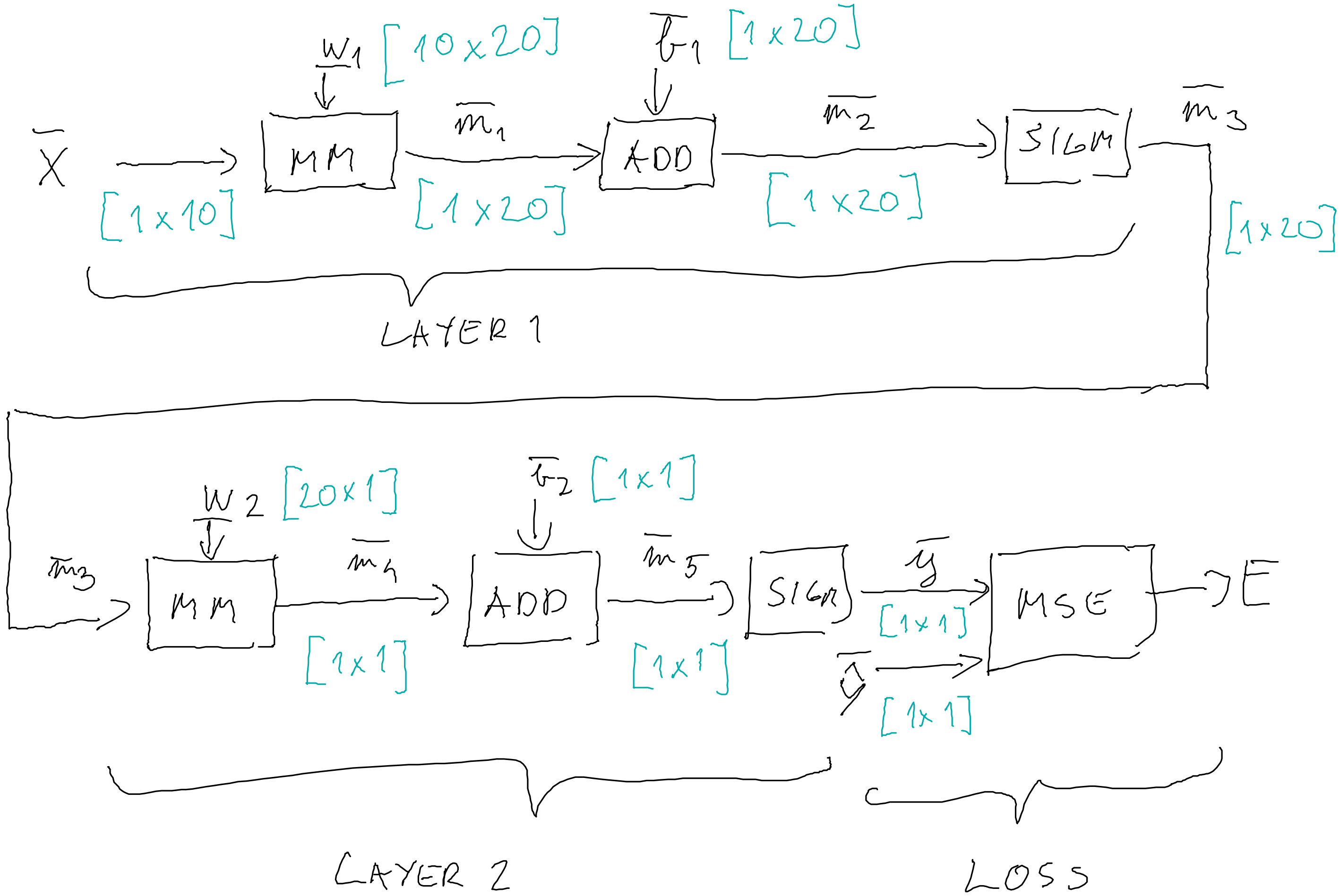
FORK

$$\boxed{\frac{\partial E}{\partial a} = \frac{\partial E}{\partial a'} + \frac{\partial E}{\partial a''}}$$

THIS PART ALWAYS EXISTS, BECAUSE WE HAVE A SINGLE, SCALAR „E".

<u>NOTE:</u> YOU WILL ENCOUNTER ALL THE VARIABLES YOU ARE INTERESTED IN DURING THE BACKWARD PASS. THEY WILL BE ON DIFFERENT BRANCHES. BUT THEIR „END" WILL BE THE SAME AND HAVE TO BE COMPUTED ONLY ONCE.

LONG EXAMPLE: 2 LAYER MLP



WE WANT TO CALCULATE:

$$\frac{\partial E}{\partial \underline{W}_1}, \quad \frac{\partial E}{\partial \bar{b}_1}, \quad \frac{\partial E}{\partial \underline{W}_2}, \quad \frac{\partial E}{\partial \bar{b}_2}$$

WHAT IS THE PATH WE HAVE TO TAKE?

$$\frac{\partial E}{\partial W_1}: \quad E \to \bar{y} \to \bar{m}_5 \to \bar{m}_4 \to \bar{m}_3 \to \bar{m}_2 \to \bar{m}_1 \to \underline{W}_1$$

$$\frac{\partial E}{\partial b_1}: \quad E \to \bar{y} \to \bar{m}_5 \to \bar{m}_4 \to \bar{m}_3 \to \bar{m}_2 \to \bar{b}_1$$

SAME: CALCULATE ONCE – DYNAMIC PROGRAMMING

START FROM THE END, MODULE BY MODULE:

$$\begin{array}{l} y \to \\ \tilde{y} \to \end{array} \boxed{\text{MSE}} \to E$$

FORWARD:

$$E = \frac{1}{2}(y - \hat{y})^2$$

BACKWARD:

$$\frac{\partial E}{\partial y} = \frac{\partial}{\partial y} \frac{1}{2}(y - \hat{y})^2 = y - \hat{y}$$

RESULT SOFAR:

$$\frac{\partial E}{\partial y} = y - \hat{y}$$

$$\bar{m}_5 \to \boxed{\text{SIGM}} \to \bar{y}$$

FORWARD:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

BACKWARD:

$$\frac{d}{dx}(1+e^{-x})^{-1} = -(1+e^{-x})^{-2} \frac{d}{dx}(1+e^{-x}) =$$

CHAIN RULE

$$= -(1+e^{-x})^{-2}\left(0 + \frac{d}{dx} e^{-x}\right) =$$

$$= -(1+e^{-x})^{-2} \underbrace{(-1)\, e^{-x}}_{\text{CHAIN RULE AGAIN}} =$$

$$= (1+e^{-x})^{-2}\, e^{-x} = \frac{e^{-x}}{(1+e^{-x})^2} =$$

$$= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x} + 1 - 1}{1+e^{-x}} =$$

$$= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x} + 1 - 1}{1+e^{-x}} =$$

$$= \frac{1}{1+e^{-x}} \cdot \left(\cancel{\frac{1+e^{-x}}{1+e^{-x}}}^{\,1} - \frac{1}{1+e^{-x}} \right) =$$

$$= \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}} \right) =$$

$$= \boxed{\sigma(x)\,(1-\sigma(x))}$$

RESULT SOFAR

$$\frac{\partial E}{\partial \overline{m_5}} = \frac{\partial E}{\partial \bar{y}} \cdot \frac{\partial \bar{y}}{\partial \overline{m_5}}$$

ORDER OF MULTIPLICATION DOESN'T MATTER. IT IS ELEMENTWISE

$$\frac{\partial E}{\partial \overline{m_5}} = \sigma(\overline{m_5})\,(1-\sigma(\overline{m_5}))\,(y-\bar{y})$$

$$\begin{tikzcd} & \bar{b}_2 \arrow[d] & \\ \bar{m}_4 \arrow[r] & \boxed{\text{ADD}} \arrow[r] & \bar{m}_5 \end{tikzcd}$$

FORWARD:

$$\bar{m}_5 = \bar{m}_4 + \bar{b}_2$$

BACKWARD:

2 OUTPUTS:

$$\frac{\partial \bar{m}_5}{\partial \bar{m}_4} = 1 \qquad \frac{\partial \bar{m}_5}{\partial \bar{b}_2} = 1$$

RESULTS SO FAR:

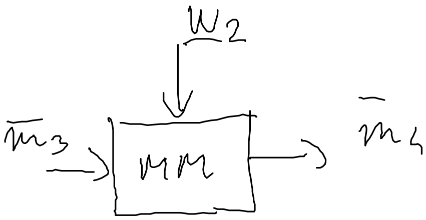
$$\frac{\partial E}{\partial \bar{b}_1} = \frac{\partial E}{\partial \bar{m}_5} \cdot \frac{\partial \bar{m}_5}{\partial \bar{b}_1} = \frac{\partial E}{\partial m_5}$$

$$\boxed{\frac{\partial E}{\partial \bar{b}_1} = \sigma(\bar{m}_5)\left(1 - \sigma(\bar{m}_5)\right)(y - \hat{y})}$$

THIS IS ALREADY ONE OF THE GRADIENTS WE ARE INTERESTED IN.

$$\frac{\partial E}{\partial \bar{m}_4} = \frac{\partial E}{\partial \bar{m}^5} \cdot \frac{\partial \bar{m}^5}{\partial \bar{m}^4} = \frac{\partial E}{\partial \bar{m}_5}$$

$$\frac{\partial E}{\partial \bar{m}_4} = \sigma(\bar{m}_5)\left(1 - \sigma(\bar{m}_5)\right)(y - \hat{y})$$



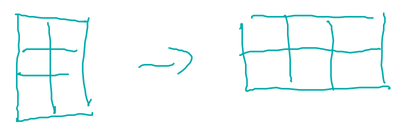
FORWARD

$$\bar{m}_4 = \bar{m}_3 \times \underline{W}_2$$

BACKWARD: 2 BRANCHES

$$\frac{\partial \bar{m}_4}{\partial \underline{W}_2} = \bar{m}_3^{T}$$

TRANSPOSE :



$$\frac{\partial \bar{m}_4}{\partial \bar{m}_3} = \underline{W}_2^{T}$$

CHAIN RULE IS NONTRIVIAL. ORDER MATTERS

$$\frac{\partial E}{\partial \underline{W}_2} = \bar{m}_3^{T} \times \frac{\partial E}{\partial \bar{m}_4}$$

MATRIX MULTIPLICATION

$\boxed{\text{ORDER MATTERS}}$

$$\frac{\partial E}{\partial \bar{m}_3} = \frac{\partial E}{\partial \bar{m}_4} \times \underline{W}_2^{T}$$

np.matmul$\left(\frac{\partial Y}{\partial \bar{m}_4}, \text{W2.T}\right)$

NOTE: THE ORDER OF MATRIX MULTIPLICATION IS DIFFERENT FOR THE 2 INPUTS.

A TRICK TO FIGURE OUT THE RIGHT ORDER IS TO WRITE DOWN AN EXAMPLE WITH A NON-SQUARE MATRIX, AND ENSURE THAT THE SHAPE OF GRADIENT MATCHES THE SHAPE OF THE INPUT.

NOTE 2: ALL OTHER OPERATIONS IN THIS EXAMPLE ARE ELEMENTWISE. EACH COMPONENT OF THE ACTIVATION VECTOR CAN BE COMPUTED SEPARATELY, AS THEY WERE SCALARS. THIS IS NOT TRUE FOR MATRIX MULTIPLICATION.

RESULTS SO FAR:

$$\frac{\partial E}{\partial \underline{W}_2} = \bar{m}_3^T \times \frac{\partial E}{\partial \bar{m}_4}$$

$$\boxed{\frac{\partial E}{\partial \underline{W}_2} = \bar{m}_3^T \times \left(\sigma(\bar{m}_5)\left(1 - \sigma(\bar{m}_5)\right)\right)(y - \hat{y})}$$

THIS IS ONE OF THE GRADS WE ARE INTERESTED IN

$$\frac{\partial E}{\partial \bar{m}_3} = \frac{\partial E}{\bar{m}_4} \times \underline{W}_2^T$$

$$\frac{\partial E}{\partial \bar{m}_3} = \left[\sigma(\bar{m}_5)\left(1 - \sigma(\bar{m}_5)\right)\left(y - \hat{y}\right) \right] \times \underline{W}_2^T$$

$\bar{m}_2 \longrightarrow \boxed{\text{SIGM}} \longrightarrow \bar{m}_3$

FROM BEFORE:

$$\frac{\partial \bar{m}_3}{\partial \bar{m}_2} = \sigma(\bar{m}_2)\left(1 - \sigma(\bar{m}_2)\right)$$

<u>RESULT SO FAR</u>

$$\frac{\partial E}{\partial \bar{m}_2} = \frac{\partial E}{\partial \bar{m}_3} \cdot \frac{\partial \bar{m}_3}{\partial \bar{m}_2}$$

$$\frac{\partial E}{\partial \bar{m}_2} = \left[\left[\sigma(\bar{m}_5)\left(1 - \sigma(\bar{m}_5)\right)\left(y - \hat{y}\right) \right] \times \underline{W}_2^T \right] \cdot \sigma(\bar{m}_2)\left(1 - \sigma(\bar{m}_2)\right)$$

FROM BEFORE

$$\frac{\partial \bar{m}_2}{\partial \bar{b}_1} = 1 \qquad \frac{\partial \bar{m}_2}{\partial \bar{m}_1} = 1$$

RESULTS SO FAR

$$\frac{\partial E}{\partial \bar{b}_1} = \frac{\partial E}{\partial \bar{m}_2} \cdot \frac{\partial \bar{m}_2}{\partial \bar{b}_1}$$

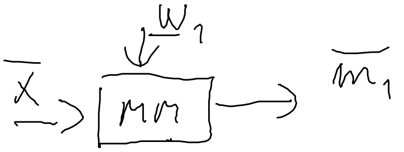
$$\boxed{\begin{aligned}\frac{\partial E}{\partial b_1} &= \Big[\big[\sigma(\bar{m}_5)(1-\sigma(\bar{m}_5))(y-\hat{y})\big] \times \underline{W}_2^T\Big] \cdot \\ &\quad \cdot \sigma(\bar{m}_2)(1-\sigma(\bar{m}_2))\end{aligned}}$$

ONE OF THE GRADIENTS WE ARE INTERESTED IN

$$\frac{\partial E}{\partial \bar{m}_1} = \frac{\partial E}{\partial \bar{m}_2} \cdot \frac{\partial \bar{m}_2}{\partial \bar{b}_1}$$

$$\frac{\partial E}{\partial \bar{m}_1} = \Big[\big[\sigma(\bar{m}_5)(1-\sigma(\bar{m}_5))(y-\hat{y})\big] \times \underline{W}_2^T\Big] \cdot$$

$$\cdot \sigma(\bar{m}_2)(1-\sigma(\bar{m}_2))$$



FROM BEFORE

$$\frac{\partial \bar{m}_1}{\partial \underline{W}_1} = \bar{x}^T$$

WE CAN IGNORE $\frac{\partial \bar{m}_1}{\partial \bar{x}}$, WE DON'T NEED GRAD FOR THE INPUTS

$$\frac{\partial E}{\partial \underline{W}_1} = \bar{x}^T \times \frac{\partial E}{\partial m_1}$$

$$\frac{\partial E}{\partial \underline{W}_1} = \bar{x}^T \times \left[\left[\left[\sigma(\bar{m}_5)\left(1 - \sigma(\bar{m}_5)\right)\left(y - \hat{y}\right) \right] \times \underline{W}_2^T \right] \cdot \sigma(\bar{m}_2)\left(1 - \sigma(\bar{m}_2)\right) \right]$$

NOTES

- THE GRADIENTS IN PRACTICE ARE NOT EXPRESSED SYMBOLICALLY AS WE DID HERE. THAT WOULD REQUIRE EACH OF THE EXPRESSIONS TO BE CALCULATED SEPARATELY - SLOW. INSTEAD THE VALUE OF THE GRADIENTS IS CALCULATED AT EACH MODULE AND PASSED TO THE NEXT. THIS WAY EACH GRAD

IS CALCULATED ONLY ONCE. THIS IS EFFICIENT.

DIFFERENCE BETWEEN GD AND BACKPROP

BACKPROPAGATION IS AN EFFICIENT DYNAMIC PROGRAMMING ALGORITHM FOR CALCULATING THE GRADIENT OF PARAMETERS WITH RESPECT TO AN OUTPUT LOSS.

GRADIENT DESCENT IS AN ALGORITHM FOR USING GRADIENTS (E.G. THE OUTPUTS OF BACKPROP) TO IMPROVE THE MODELS PERFORMANCE (DECREASING THE LOSS).
IT IS A WAY TO CHANGE THE MODELS PARAMETERS BASED ON THEIR GRADIENTS.

TRAINING TRICKS

MOMENTUM

SGD IS VERY SLOW. AN EASY WAY TO IMPROVE IT IS THE MOMENTUM.

MOMENTUM OPTIMIZER ACCUMULATES THE GRADIENT AND DOES THE PARAMETER UPDATE WITH THE ACCUMULATED GRADS. THIS IS LIKE MOMENTUM IN PHYSICS.

WEIGHTS ARE INDEPENDENT IN PERSPECTIVE OF THE OPTIMIZER. i-th WEIGHT IS w_i

$$v_i^{NEW} = \mu \, v_i^{OLD} + \frac{\partial E}{\partial w_i}$$

$$w_i^{NEW} = w_i - \eta \cdot \bar{v}_i$$

– IN SGD THIS TERM IS $\frac{\partial E}{\partial w_i}$

LEARNING RATE (TYPICALLY 0.1 ... 0.0001)

μ – MOMENTUM (TYPICALLY 0.9 .. 0.99)

INTUITION: IF GRADIENT CONSISTENTLY POINTS TO A DIRECTION, WE CAN ACCELERATE IN THAT DIRECTION (THE DIRECTION IS UNLIKELY TO CHANGE). HOWEVER IF THE GRADIENT CHANGES DIRECTION FREQUENTLY, WE SHOULD GO SLOWER IN ORDER NOT TO OSCILLATE NEAR THE MINIMUM.

SIGN IS CONSISTENT, ACCELERATE!

THE GRAD CHANGES DIRECTION SLOW DOWN!

WEIGHT DECAY

COMMONLY USED REGULARIZER. PENALIZES THE MAGNITUDE OF THE WEIGHTS.

INTUITION: MAKE THE UNNECESSARY WEIGHTS 0. (IF $E[GRAD] = 0$, THERE IS NO DRIVING FORCE FOR MAKING IT NONZERO, BUT THE WD DRIVES THEM TOWARDS ZERO)

LETS CALL THE UPDATE Δ. IN CASE OF SGD, $\Delta = \frac{\partial E}{\partial w_i}$, IN CASE OF MOMENTUM, $\Delta = v_i$.

$$\bar{w}_i^{NEW} = \bar{w}_i^{OLD} - \eta \left(\Delta + \gamma \, \bar{w}_i^{OLD} \right)$$

WEIGHT DECAY $\gamma \sim 1e^{-4} \ldots 1e^{-8}$

CAN BE REORGANIZED:

$$\bar{w}_i^{NEW} = (1 - \eta\gamma)\, \bar{w}_i^{OLD} - \eta \Delta$$

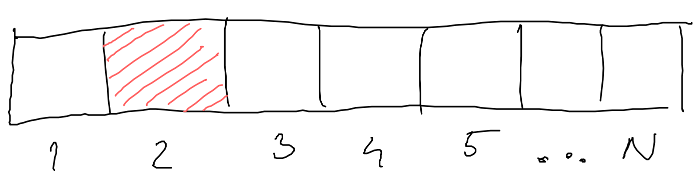
CROSS VALIDATION

IN CASE THE AMOUNT OF DATA AVAILABLE IS LOW, SPLITTING THE DATASET INTO TRAIN / VALIDATION / TEST SETS MIGHT BE A PROBLEM (TOO MUCH DATA IS WASTED)

CROSS VALIDATION AVOIDS EXCLUDING THE VALIDATION DATA FROM TRAINING, BUT THE MODEL SHOULD BE TRAINED MULTIPLE TIMES. (TAKES MUCH LONGER).

N - WAY CROSS VALIDATION

1. DIVIDE THE TRAINING SET TO N CHUNKS



2. TRAIN N DIFFERENT MODELS:

   FOR iTH MODEL, EXCLUDE ith BLOCK FROM TRAINING, AND USE IT FOR VALIDATION. IF THERE ARE M SAMPLES, YOU WILL TRAIN ON $\frac{N-1}{N} M$ AND VALIDATE ON $\frac{M}{N}$ OF THEM.

(3) CALCULATE YOUR VALIDATION ACCURACY BY AVERAGING THE ACCURACIES OF ALL N MODELS.

(4) SELECT HYPERPARAMETERS BASED ON THE AVERAGE PERFORMANCE

(5) RETRAIN THE MODEL ON FULL TRAINING DATA WITH THE SELECTED HYPER-PARAMETERS.