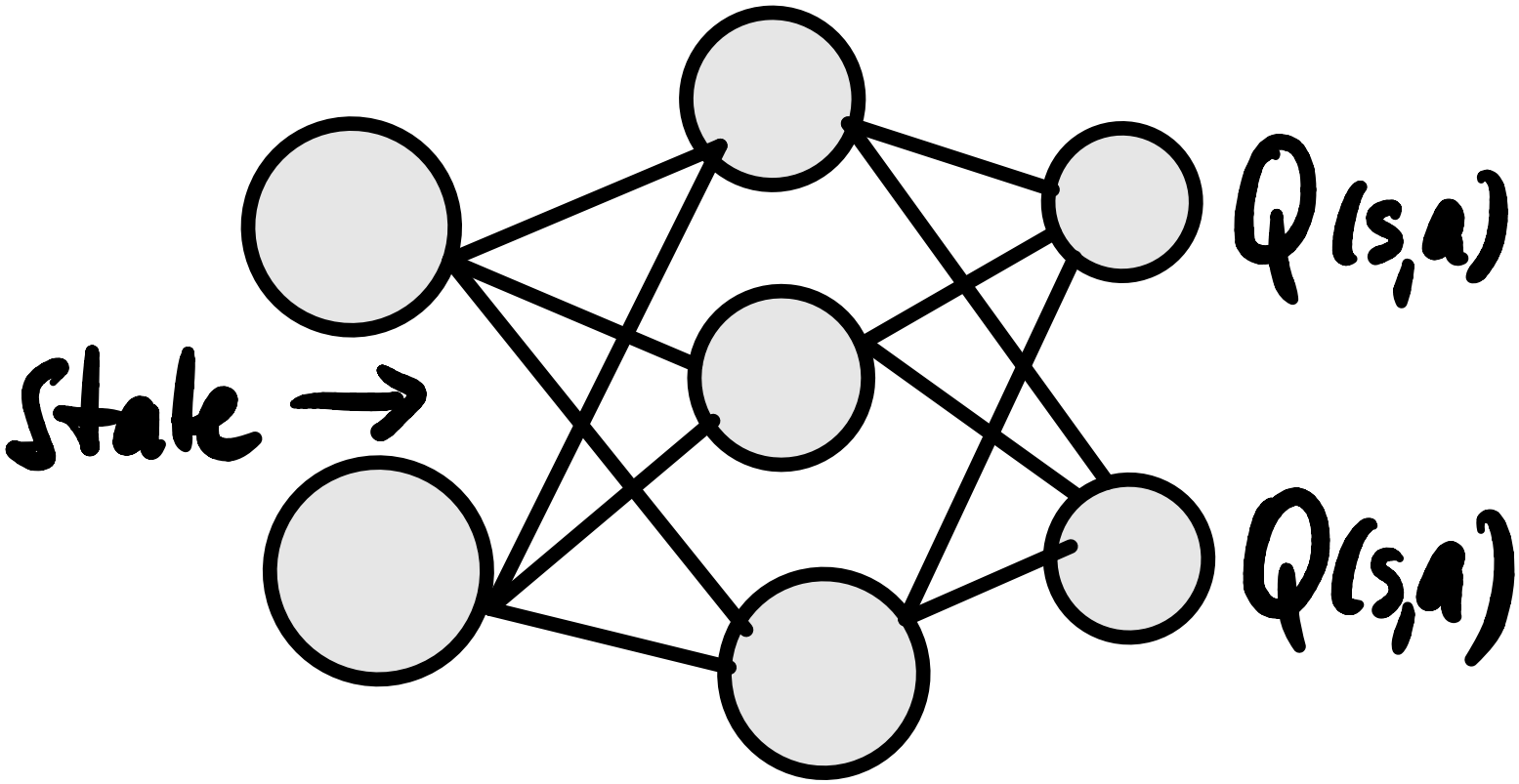
REINFORCE

09.02.2021

Value-Based Methods

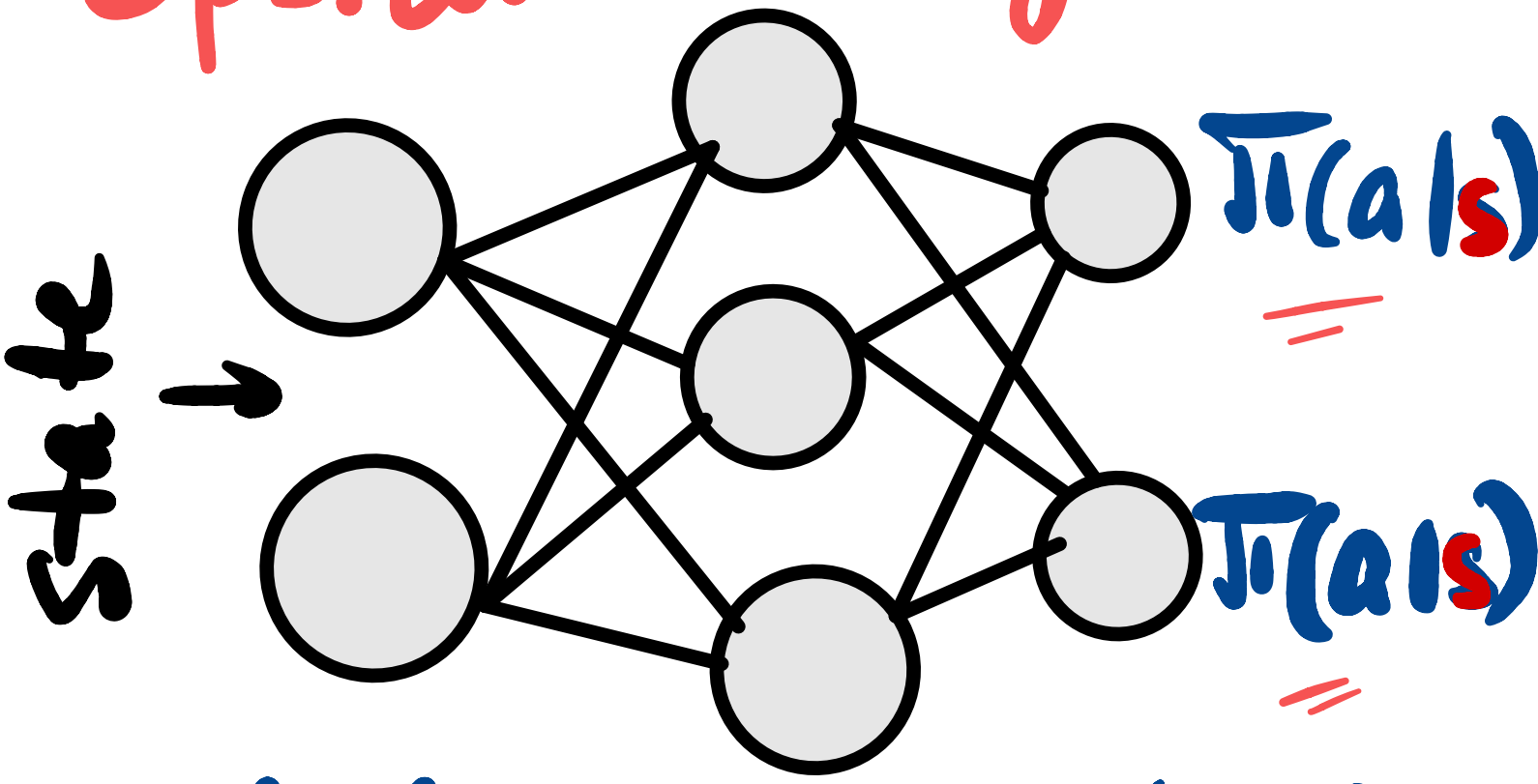


$$a^* = \arg\max_a Q(s,a)$$

Policy: action that gives **the best** value

Policy Based

Epsilon greedy ?



Probability we should perform the action given the input state

DQN – discrete action space

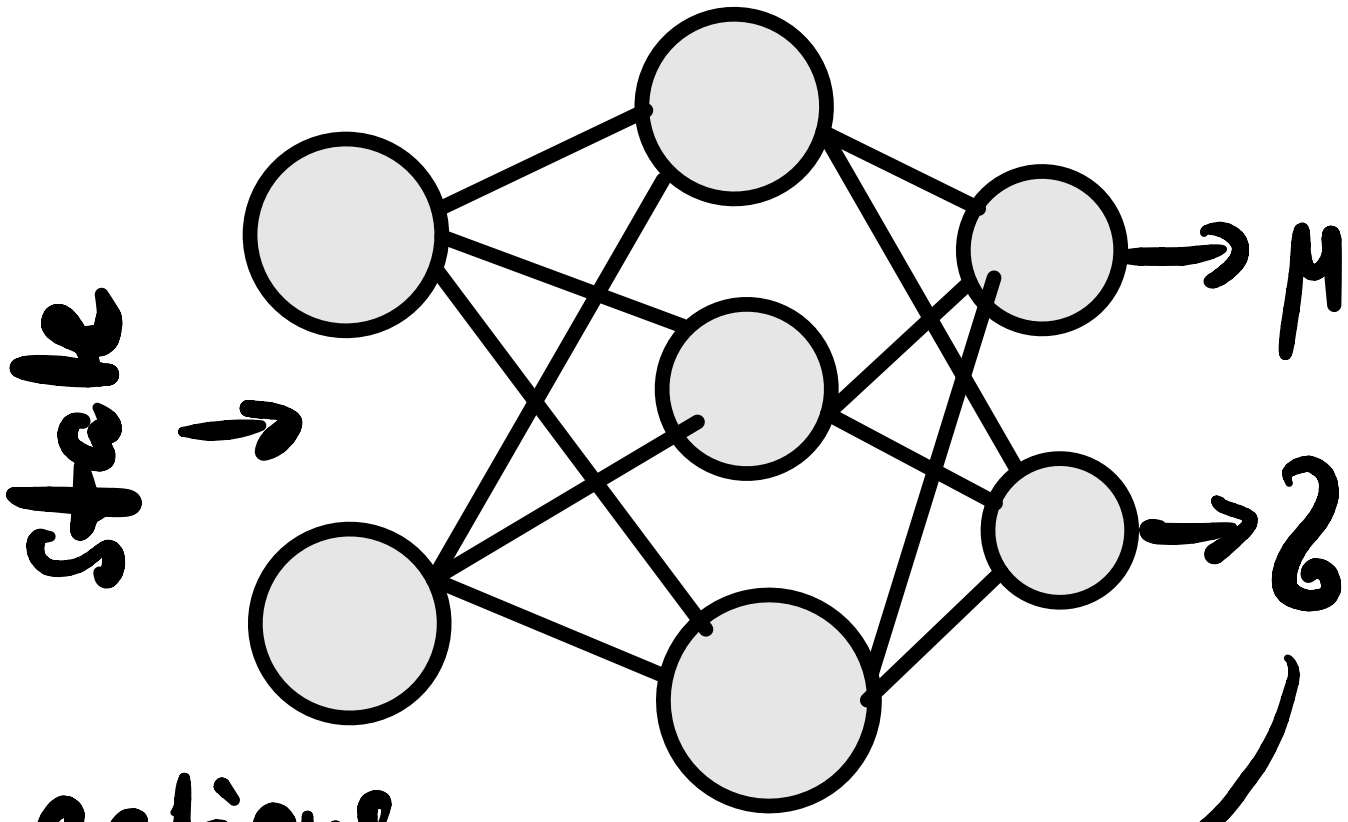
each output – action

Policy Networks –

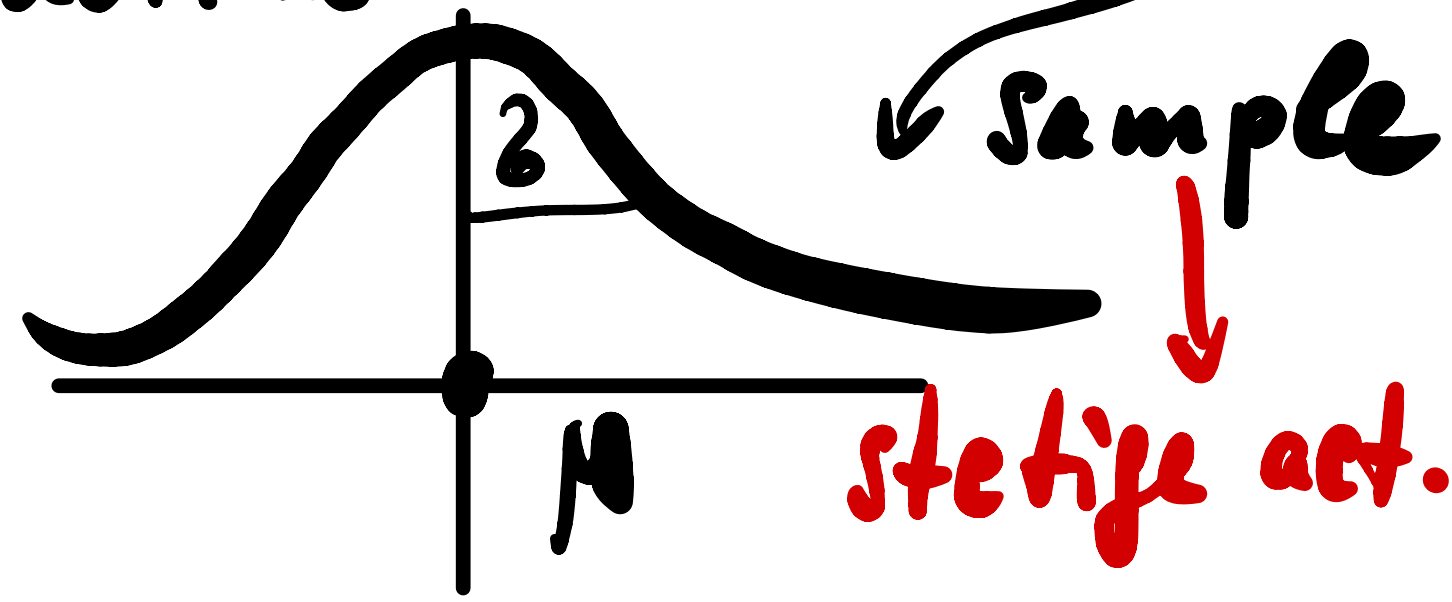
can output parameters of any distribution

e.g. μ and σ

Policy



actions



sample

stetige act.

Training a policy

DQN:

$$E[(G - Q(s,a))^2]$$

dqn output

return

$\rightarrow 0$

and

Kommt aus NN

$\pi_\theta(a|s)$?

mit was vergleich.

Q-Values bzw.

$$G = \frac{R_0 + \gamma^1 \cdot R_1 + \dots \gamma^n R_n}{}$$

aus dem Spiel

0.8

Zahl

$\pi_\theta(a|s)$ ~ G

100

θ - gewichte des NNs

$$\left(f(G) - \pi\right)^2 \overset{!}{=} \text{min}$$

Policy Gradient Derivation

1. max $\sum_i p \cdot R_i$

$G = E[R_1 + R_2 + \dots R_T]$

return → exp. value

2. stak-action pairs

$\Omega = \{(s_1, a_2), \dots (s_T, a_T)\}$

3. $G_{(\Omega)} = \sum_t R(s_t, a_t)$

return

Gradient ASCENT

ascent

$$\theta_{new} = \theta + \nabla J_\theta \cdot LR$$

LR – Learning Rate

$$\nabla J_\theta \rightarrow$$

$$\theta_{new} = \theta_{old} - (\ominus \nabla J_\theta \cdot LR)$$

4. Objective $J(\theta)$

$$J(\theta) = \sum_{\Omega} P(\Omega;\theta) \cdot G(\Omega)$$

Probs

Network params

Generische

Policy which want to find

Returns

Summe über State-actions

We want to maximize our objective Funktion

maximize expected return!

5. Objective Gradient

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_\Omega P(\Omega;\theta) \cdot G(\Omega)$$

$$\frac{\partial J}{\partial \theta}(\Omega)$$

$$\nabla_\theta J(\theta) = \sum_\Omega \nabla_\theta P(\Omega;\theta) \cdot G(\Omega)$$

$$(a+b)' = a' + b'$$

$$\nabla(a+b) = \nabla a + \nabla b$$

6. Trick :

$$\nabla_\theta J(\theta) = \sum_\Omega P(\Omega;\theta) \cdot \frac{\nabla P(\Omega;\theta)}{P(\Omega;\theta)} \cdot G(\Omega)$$

$$\nabla \log f = \frac{\nabla f}{f}$$

$$\nabla_\theta J(\theta) = \sum_\Omega P(\Omega;\theta) \nabla_\theta \log P(\Omega;\theta) \cdot G(\Omega)$$

Summe

probs

Erwartungswert

$$\ln(x)' = \frac{1}{x} = \frac{f'}{f} \qquad f = x$$

$$\mathbb{E}[\,\cdot\,]$$

$$\nabla_\theta J(\theta) = E\left[\nabla_\theta \log P(\Omega;\theta) \cdot G(\Omega)\right]$$

1. Gradient tries to make P more probable
2. IF G is positive, reinforce the actions that led to it

$$G = p(r + \gamma \max_a Q)$$

BEQ

direct

nur diskret actionen

Value Iter

SARS(A)

(Tabular Q-L.)

DQN

DDQN

RDQN

} Paper 2013

Value based

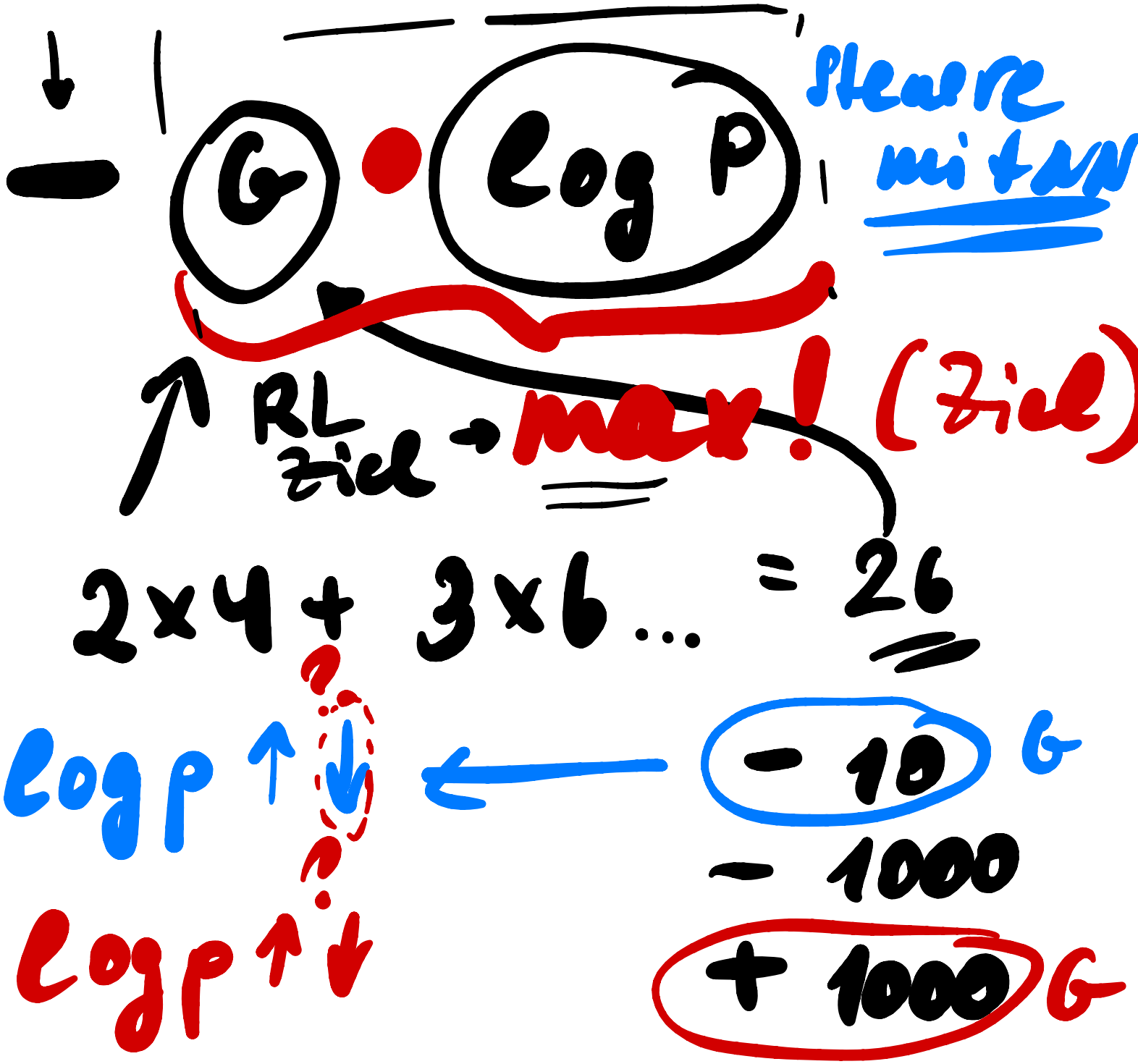
--->

direkt policy ausrechnen

$\pi(a|s)$

Familie von

Policy Gradient Verfahren


Steuer mit AN
G
log P
RL Ziel → max! (Ziel)
2x4 + 3x6 ... = 26
log p ↑↓
log p ↑↓
- 10 G
- 1000
+ 1000 G

NN → Diener

RL – wir

wir: $G \cdot \log p \uparrow$

max

NN: kann nur minimieren

$-\uparrow G \cdot \log p_\theta \downarrow$

REINFORCE actions mit G + re

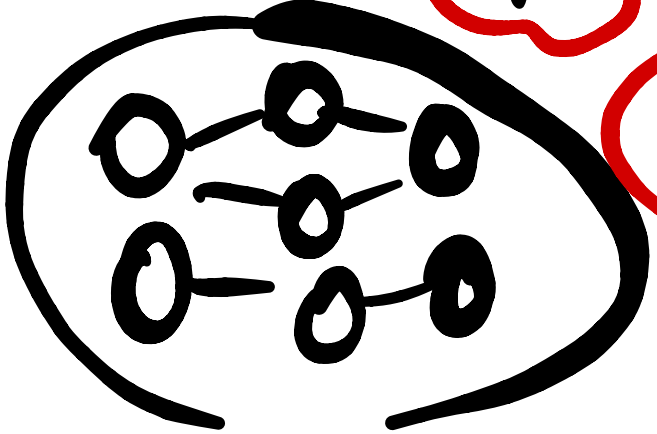
Spoiler :)

$-\log p \cdot G$ A2C

A

Stalt Advantage

$A = Q_{s,a}$ (Neu) $- V_s$ (Schon da)



Kritiker

A2C

Actor (REINF.)

Critic (DQN)

log p · A (state)

A = Q(neu) − V(alt)

(vorgesch. von Actor)

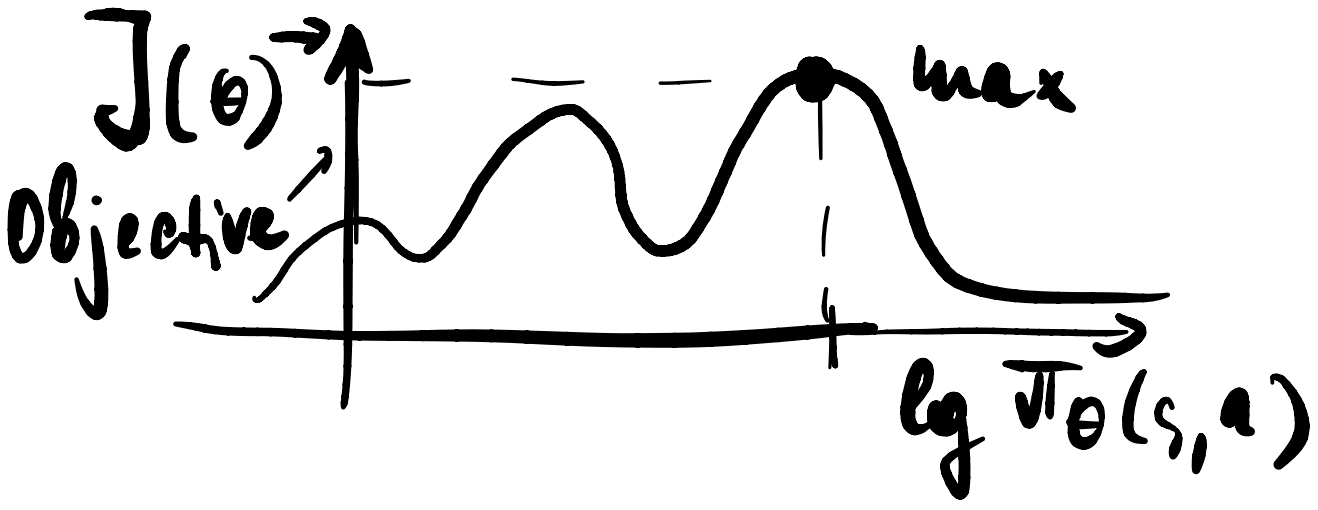
$$L(\theta) = -\log P(\Omega;\theta) \cdot G(\Omega)$$

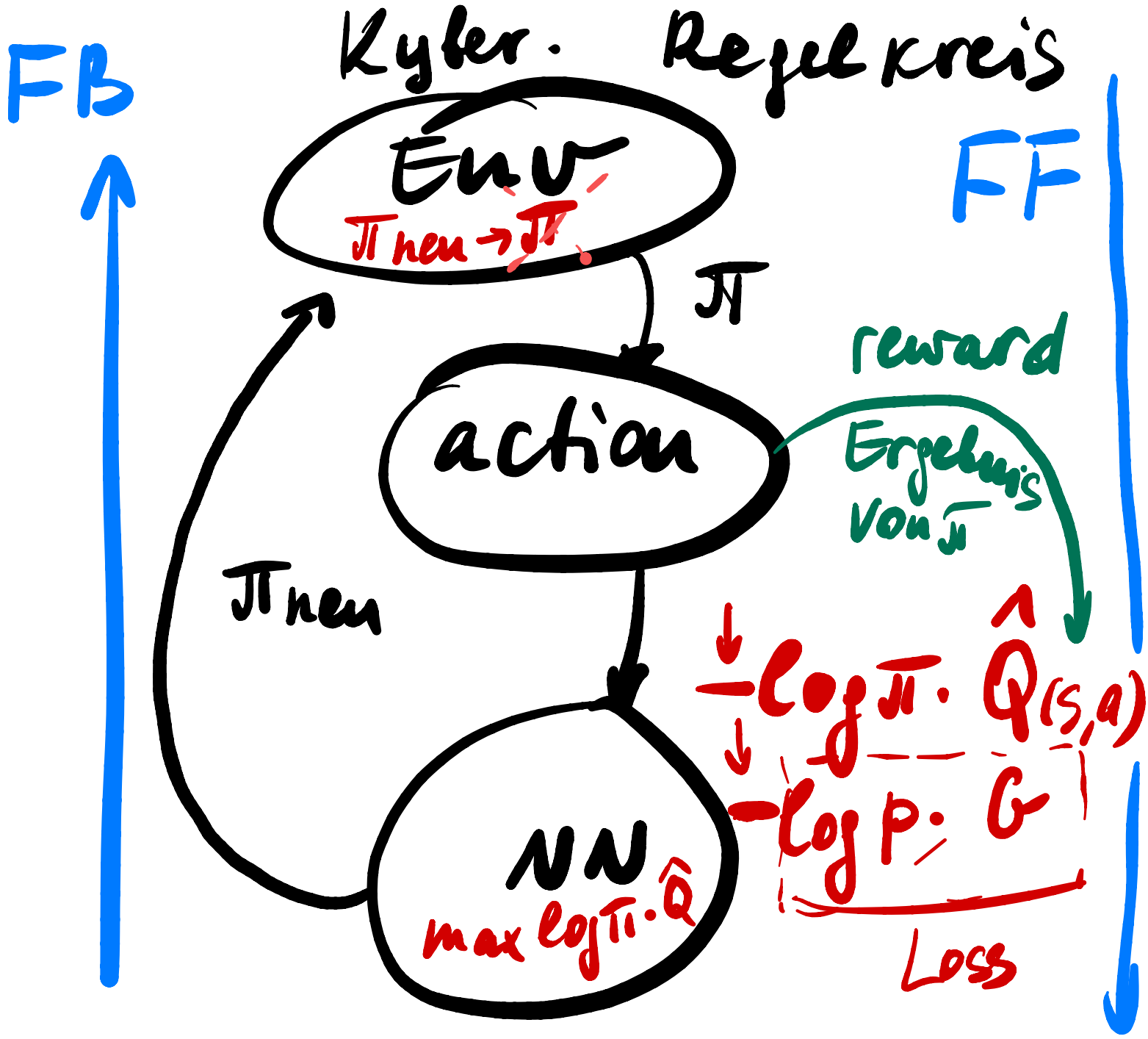
Objective => loss

By setting it negative we perform Gradient ascent, looking for min. of the negative Loss

Policy Grad.

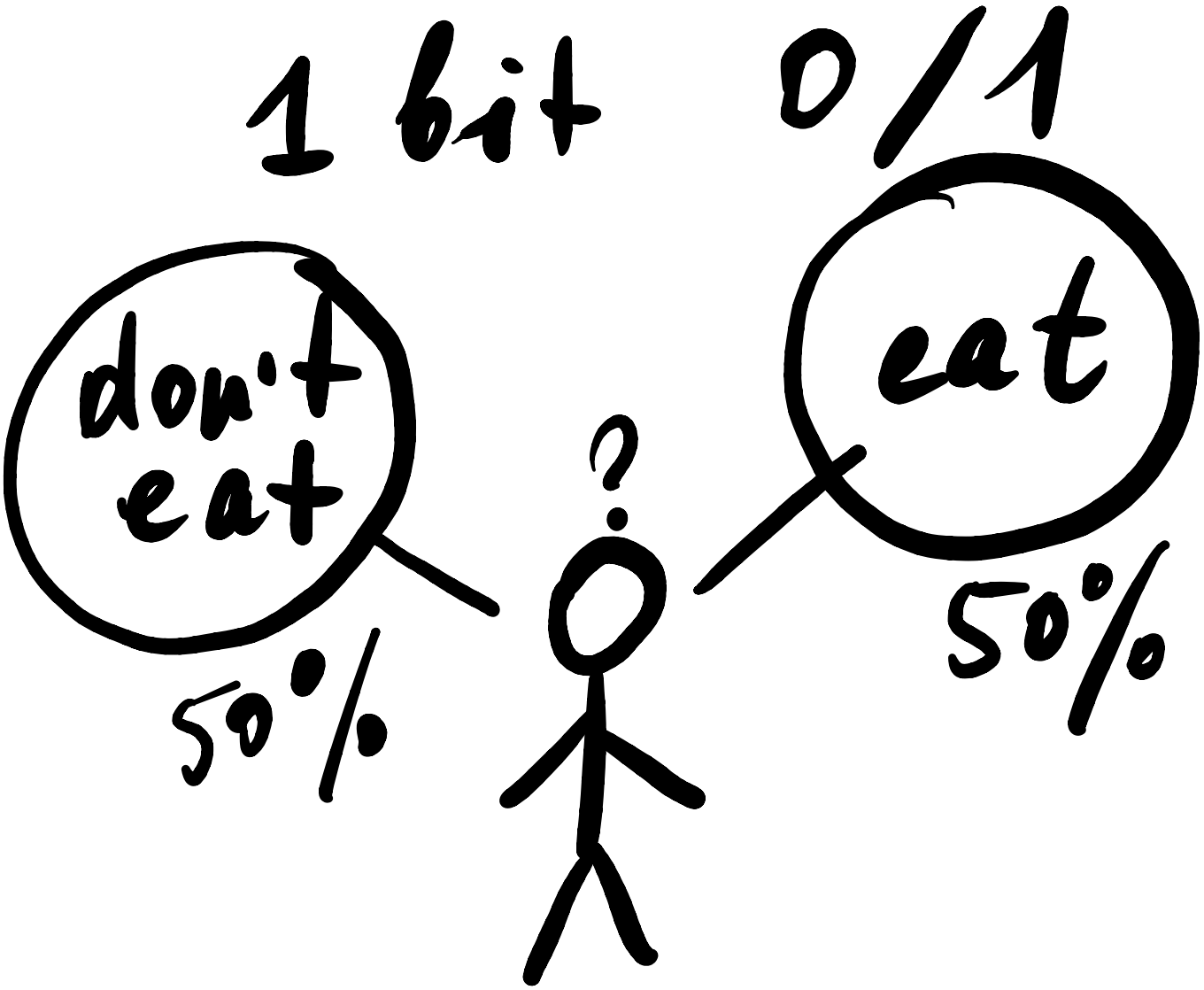
$$\theta_{t+1} = \theta_t - (-\nabla \log \pi_\theta(s,a) \cdot \hat{Q}(s,a))$$
$$= \theta_t + \nabla \log \pi_\theta(s,a) \cdot \hat{Q}(s,a)$$




Kyber. Regelkreis
FB
FF
Env
π neu → π
π
reward
Ergebnis von π
action
π neu
$-\log\pi \cdot \hat{Q}(s,a)$
$-\log P \cdot G$
Loss
NN
max $\log\pi \cdot \hat{Q}$

Claude Shannon

1 bit 0/1



→ 0 DE
1 E

Unsicherheit um Faktor
(2) reduziert

1. Entropy
2. CrossEntropy
   ↳ Objective classification
3. KL Divergence
   ↳ Autoencoder

0000
0000

↓

gleich
wahrsch.

$2^3 = 8$

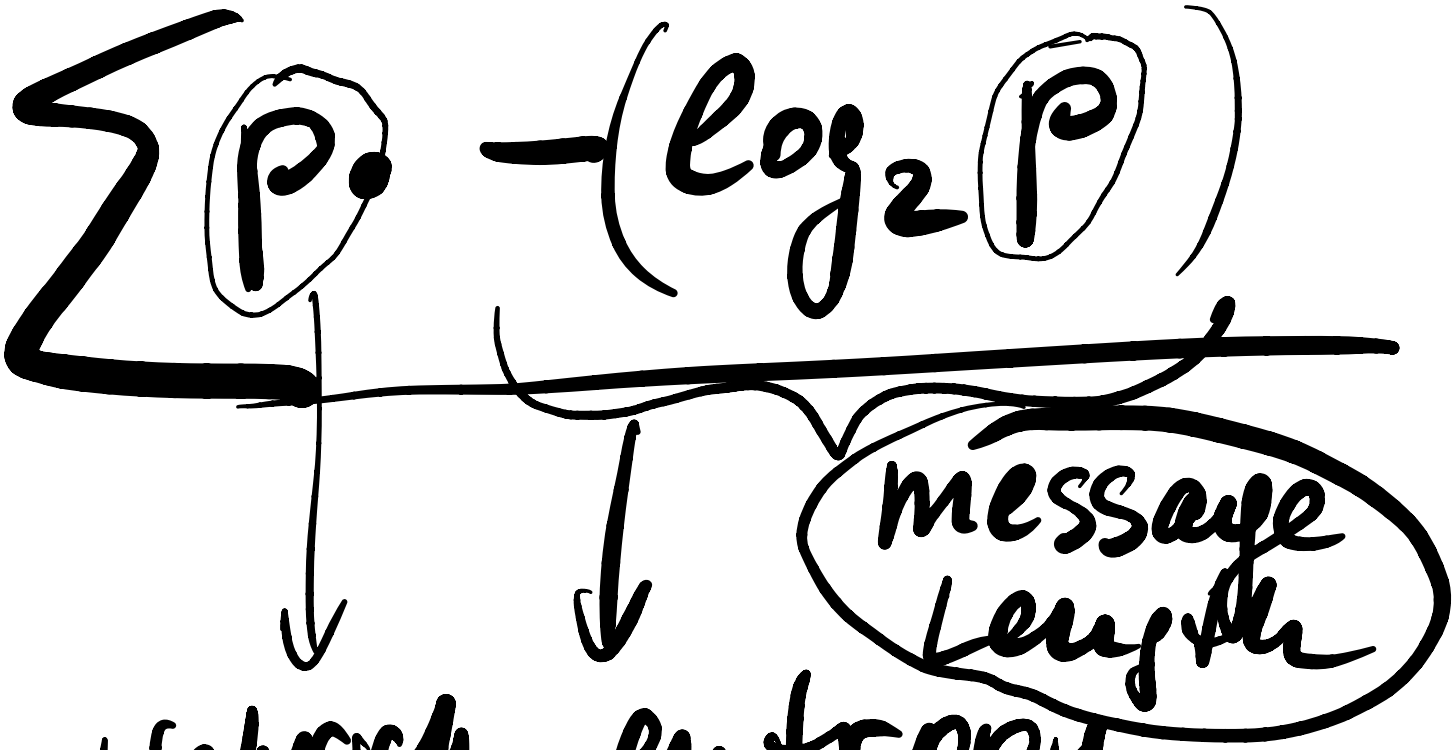
3 bit

$\log_2 8 = 3$

Message
Length

$-\log_2 \frac{1}{8} = \log_2 8$

p



Wahrsch. entropy

je komplexer
desto länger message
um das zu beschreiben

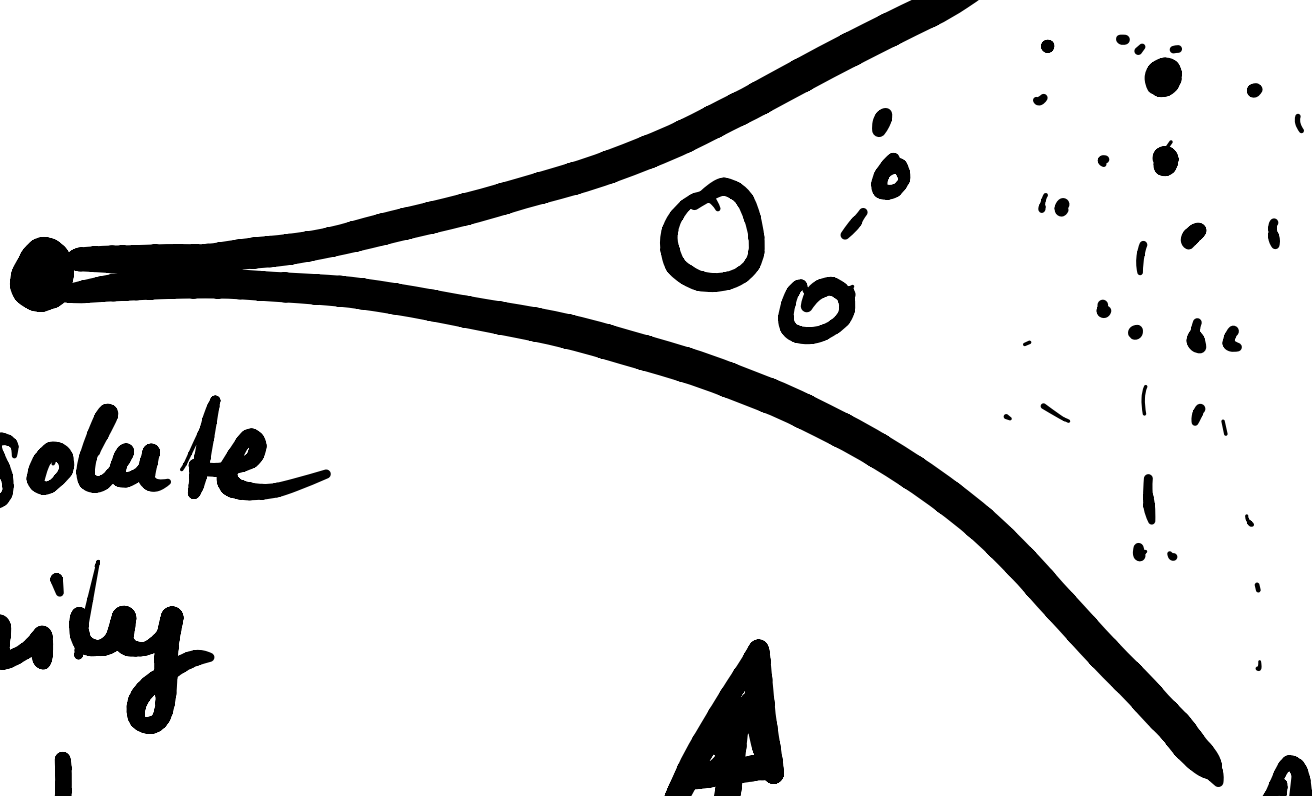
$p \rightarrow \frac{1}{p}$

$10.000 \quad \frac{1}{10.000}$

$-\log \frac{1}{10.000} \approx +\text{grosso}$

$\lim -\log p \rightarrow +\infty$

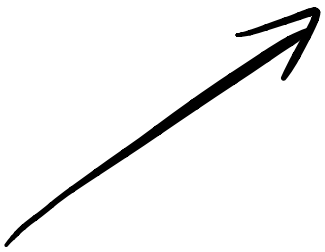
$p \rightarrow 0$



Absolute
Unity
Entropy = 0 ↑ Entropy ↑

message ↑

DNS →



0 0

75% 25%

$-\ln 0.75$ $-\ln 0.25$

message message

length

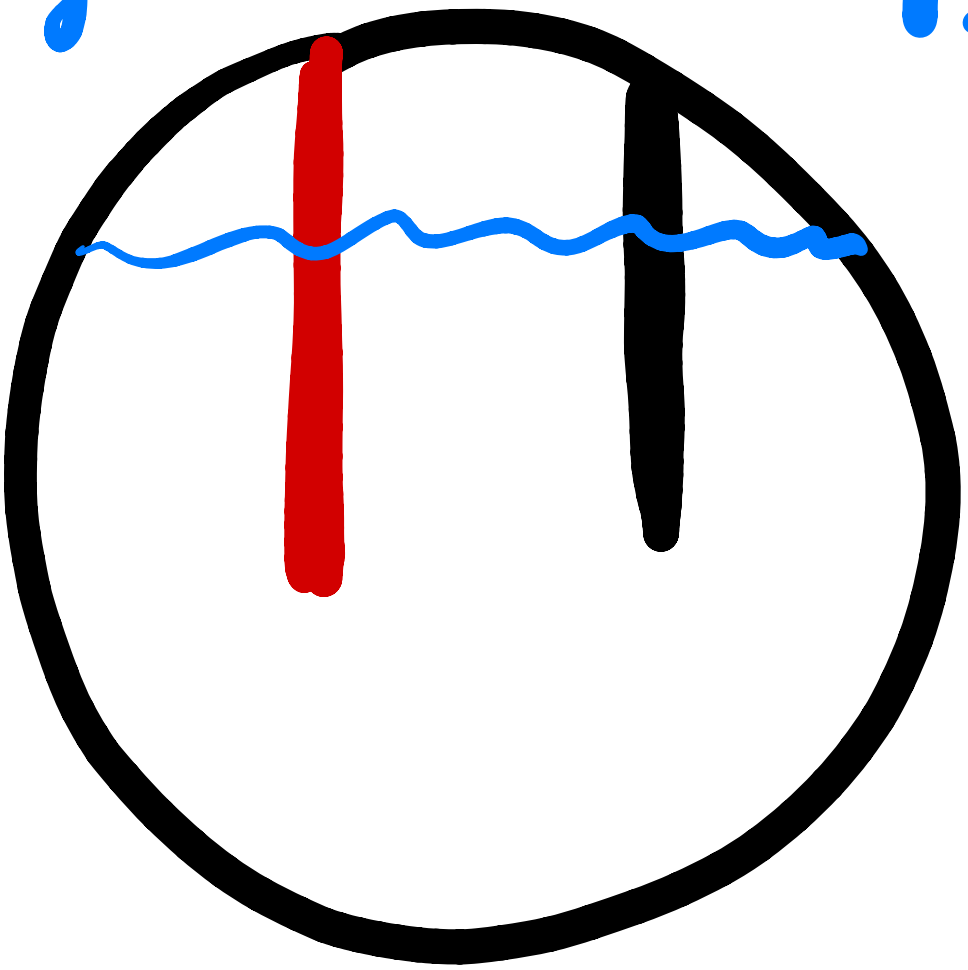
$0.75 \cdot \ln \frac{1}{0.75} + 0.25 \ln \frac{1}{0.25}$

Number Bits Bits

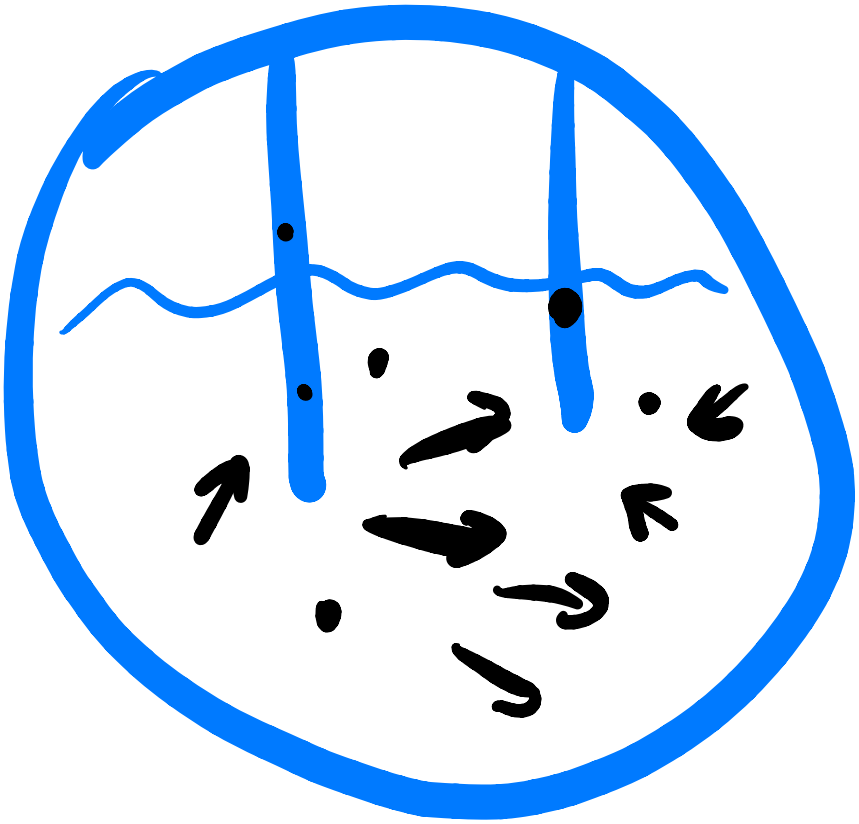
Erwartungswert Mess-Länge

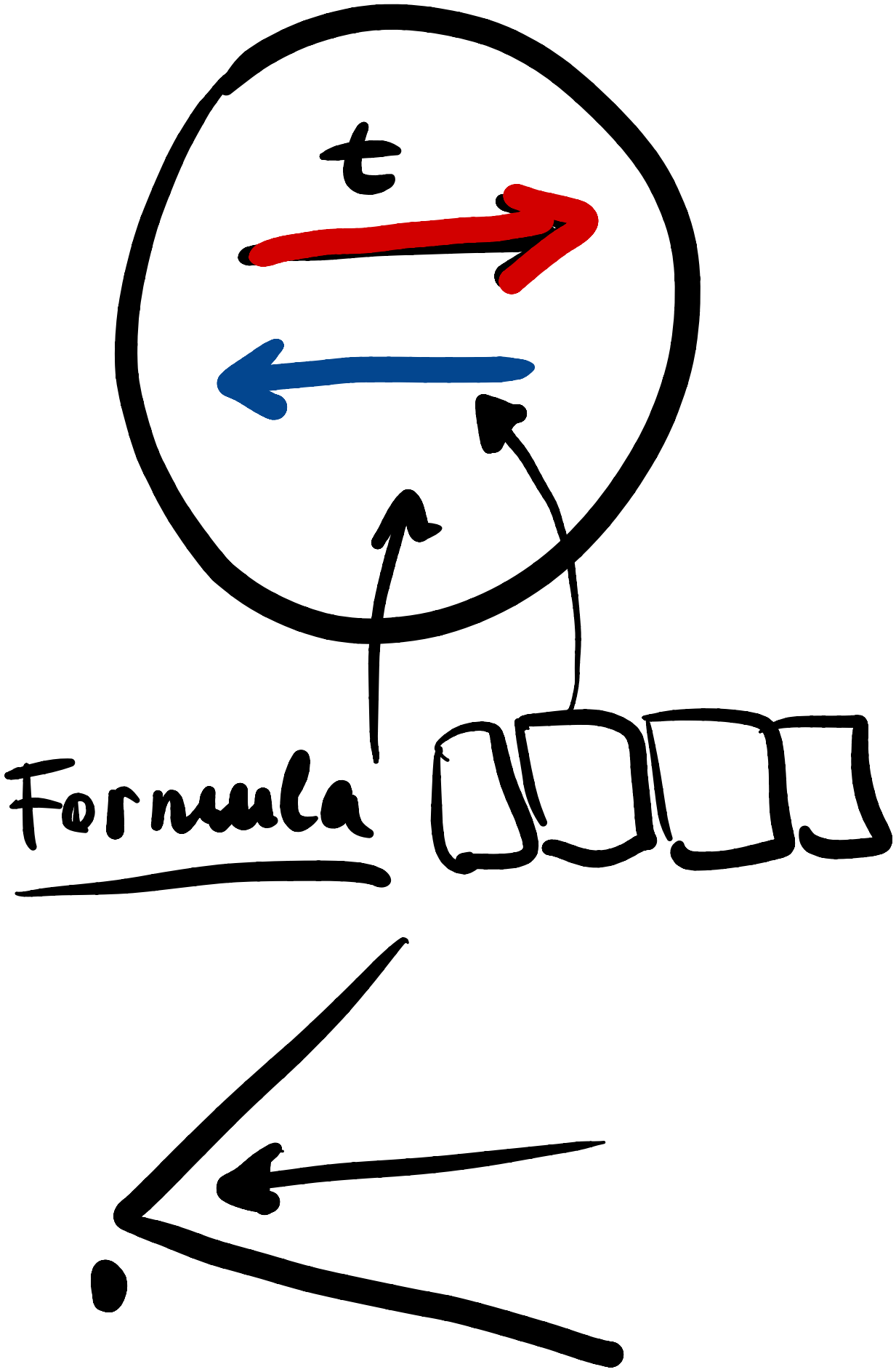
Physik

1.



2.




t
Formula

2. Cross Entropy

$$-\sum p \cdot \ln p$$

Wahre Komplexität

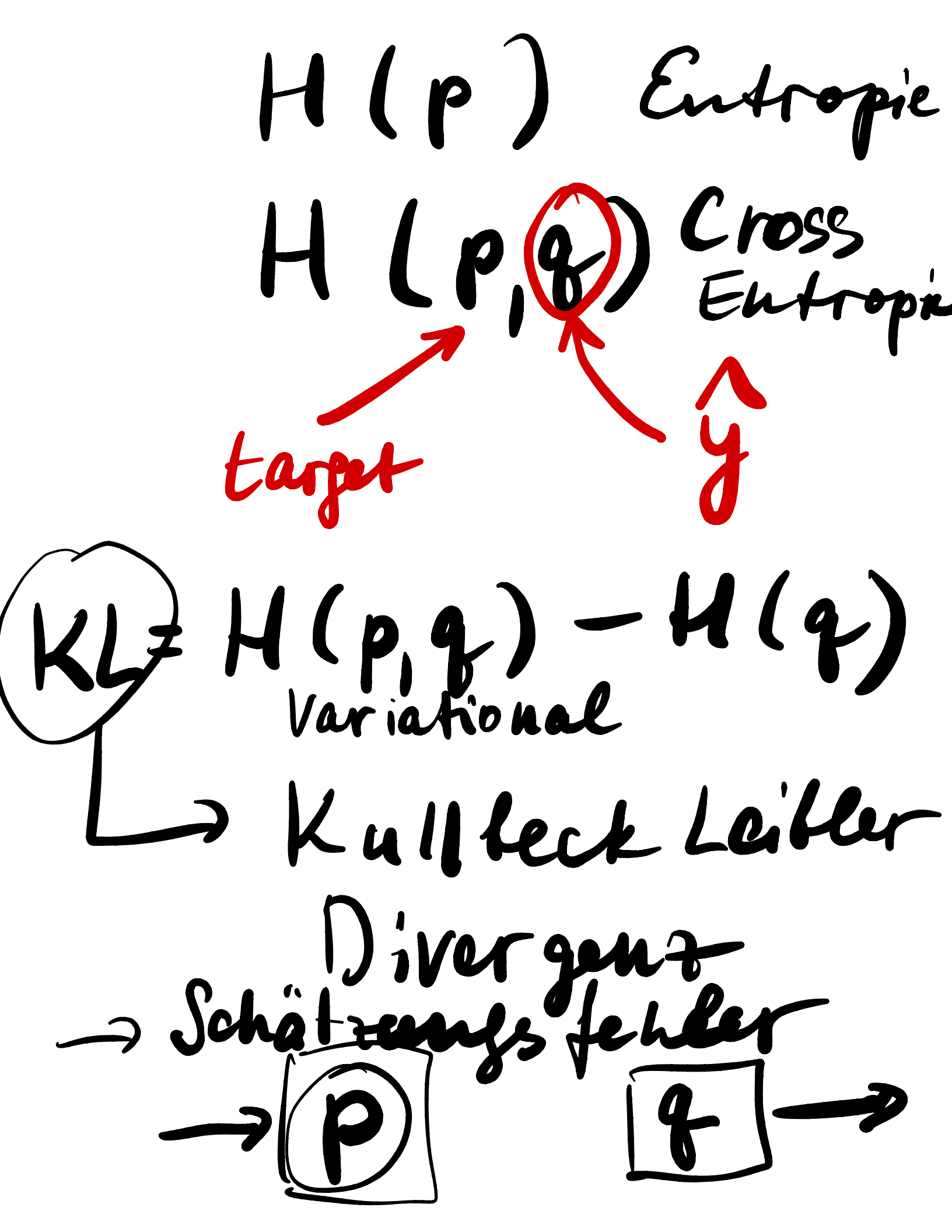
$$-\sum p \cdot \ln q$$

(2) (1)

↗ Cross Entropy

$\ln q$ – unsere schätz. von Message.

– Komplexität


H(p) Entropie
H(p,q) Cross Entropie
target
ŷ
KL = H(p,q) − H(q)
Variational
Kullbeck Leibler
Divergenz
→ Schätzungsfehler
p
q

VAE

Din ≠ Dout

Verteilung p = q Verteilung

Stat. Properties

↳ H(p,q) = H(p)

↳ KL = 0 !

Supervised

Obj. Cross Entr.

Reale Entropie

Geschätzte Kompl.

$H(p,q) = H(p)$

$KL = 0$

O

Eat

90%

O

DE

10%

$$-(0.9 \cdot \log 0.9 + 0.1 \log 0.1)$$

0.5 0.5

$$-(0.9 \cdot \log 0.5 + 0.1 \log 0.5)$$

$E = mc^2$

$H(p,q) > H(p)$