C] Restricted Boltzmann Machine

The Hamiltonian is given by

$$\mathcal{H}[\vec{s},\vec{\tau}] = -\sum_{i,a} s_i W_{ia} \tau_a - \sum_{i=1}^{N_v} \theta_i s_i - \sum_{a=1}^{N_h} \eta_a \tau_a$$

$\vec{s}$: visible nodes $\{0,1\}$

$\vec{\tau}$: hidden nodes $\{0,1\}$

θ_i: vis biases (magn fields)

η_a: hid biases (magn fields)

$$p(\vec{s},\vec{\tau}) = \frac{1}{Z} e^{-\mathcal{H}[\vec{s},\vec{\tau}]} \quad \text{with} \quad Z = \sum_{\{\vec{s}\},\{\vec{\tau}\}} \exp(-\mathcal{H}[\vec{s},\vec{\tau}])$$

Why RBM can be better than Hopfield model?

1) it does contain patterns : $\boxed{W_{ia}}$

ii) → it can modelize higher-order correlations

Let's consider $\vec{\eta} = \vec{\theta} = 0$ (for simplicity)

$$p(\vec{s}) = \frac{1}{Z}\sum_{\{\tau\}} \exp\Big(\sum_{i,a} s_i W_{ia} \tau_a\Big) = \frac{1}{Z}\prod_a \sum_{\tau_a=0,1} e^{\tau_a\left(\sum_i W_{ia} s_i\right)}$$

$$= \frac{1}{Z}\prod_a\Big(1 + e^{\sum_i W_{ia} s_i}\Big) = \frac{1}{Z}\exp\Big[\sum_a \log\Big(1 + e^{\sum_i W_{ia} s_i}\Big)\Big]$$

W_{ia} small

$$\simeq \frac{1}{Z}\exp\Big[\sum_a \log\Big(1 + 1 + \sum_i W_{ia} s_i + \frac{1}{2}\Big(\sum_i W_{ia} s_i\Big)^2 + \frac{1}{3!}\Big(\sum_i W_{ia} s_i\Big)^3 + \dots\Big)\Big]$$

$O(w)$

$$\simeq \sum_{\tau}^{N_h} \exp\left[\frac{1}{2}\sum_{a,i} W_{ia} S_i \right.$$

$O(w)$

$$+ \frac{1}{4}\sum_{i,j} s_i\left(\sum_a W_{ia} W_{ja}\right)s_j - \frac{1}{8}\,(\ldots)$$

$O(w^2)$

$$+ \frac{1}{3!\,2}\sum_{i,j,h} s_i s_j s_h\left(\sum_a W_{ia} W_{ja} W_{ha}\right) + \ldots$$

$O(w^3)$

$+$

1) Sampling with RBM

Sampling here, is as simple as for the Hopfield model!

First we need to compute the cdt^d dist

$$p(s_i \mid \vec{\tau}) = \frac{e^{s_i\left(\sum_a W_{ia}\tau_a + \theta_i\right)}}{\sum_{s_i} e^{s_i\left(\sum_a W_{ia}\tau_a + \theta_i\right)}} = \frac{e^{s_i\left(\sum_a W_{ia}\tau_a + \theta_i\right)}}{1 + e^{\sum_a W_{ia}\tau_a + \theta_i}}$$

$$p(s_i = 1 \mid \vec{\tau}) = \frac{1}{1 + e^{-\sum_a W_{ia}\tau_a - \theta_i}} = \mathrm{sig}\left(\sum_a W_{ia}\tau_a + \theta_i\right) = \langle s_i \rangle_{|\vec{\tau}}$$

where : $\mathrm{sig}(x) = \dfrac{1}{1 + e^{-x}}$. sigmoid fct.

By symmetry, we have

$$p(\tau_a = 1 \mid \vec{s}) = \mathrm{sig}\left(\sum_i W_{ia} S_i + \eta_a\right) = \langle \tau_a \rangle_{|\vec{s}}$$

Again we can sample configurations layer-wise:

1) we start with a rdm config: $\vec{s}^0 \sim$ bernoulli $(p=0.5)$

ii) $\vec{\tau}^0 \sim p(\tau_a | \vec{s}^0) \sim$ bernoulli$\left(1-p_a = \operatorname{sig}\left(\sum_i w_{ia} s_i^0 + \eta_a\right)\right)$

$\vec{s}^1 \sim p(s_i | \vec{\tau}^0) \qquad \vec{\tau}^1 \sim p(\tau_a | \vec{s}^1)$

$\vec{s}^2 \sim p(s_i | \vec{\tau}^1) \quad \dots$

$\vdots$

$\{\vec{s}^T, \vec{\tau}^T\}$

II) Learning of RBM: the procedure is as always:

we need to maximize the likelihood

Let's consider a dataset: $\{\vec{s}^{(m)}\}_{m=1,\dots,M}$

the log-like is $\mathcal{L} = \frac{1}{M}\sum_m \log\left[p(\vec{s}^{(m)})\right] = \frac{1}{M}\sum_m \log\left[\sum_{\{\tau\}} p(\vec{s}^{(m)}, \vec{\tau})\right]$

$$\mathcal{L} = \frac{1}{M}\sum_m \log\left[\sum_{\{\tau\}} \exp\left(\sum_{ia} s_i w_{ia} \tau_a + \underbrace{\sum_i \theta_i s_i^{(m)}} + \sum_a \eta_a \tau_a\right)\right] - \log Z$$

$\langle s_i \rangle_D =:$

$$= \frac{1}{M}\sum_m \sum_i \theta_i s_i^{(m)} + \frac{1}{M}\sum_m \log\left(\prod_a \sum_{\tau_a=0,1} e^{\tau_a\left(\sum_i w_{ia} s_i^{(m)} + \eta_a\right)}\right) - \log Z$$

$$\langle f(\vec{s},\vec{\tau})\rangle_D := \frac{1}{M}\sum_m \sum_{\{\tau\}} f(\vec{s}^{(m)},\tau)\cdot p(\vec{\tau}\,|\,\vec{s}^{(m)})$$

$$= \sum_i \theta_i \langle s_i\rangle_D + \frac{1}{M}\sum_m\left[\sum_\alpha \lg\left(1+e^{\left(\sum_i w_{i\alpha} s_i^{(m)}+\eta_\alpha\right)}\right)\right] - \lg Z$$

We compute the gradients.

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \langle s_i\rangle_D - \underbrace{\frac{1}{Z}\sum_{\{s,\tau\}} s_i\, e^{-\mathcal{H}[\vec{s},\vec{\tau}]}}_{:=\langle s_i\rangle_{RBM}} = \langle s_i\rangle_D - \langle s_i\rangle_{RBM}$$

$$\frac{\partial \mathcal{L}}{\partial \eta_\alpha} = \frac{1}{M}\sum_m \frac{e^{\sum_i w_{i\alpha} s_i^{(m)}+\eta_\alpha}}{1+e^{\sum_i w_{i\alpha} s_i^{(m)}+\eta_\alpha}} - \langle \tau_\alpha\rangle_{RBM}$$

$$= \frac{1}{M}\sum_m \underbrace{p(\tau_\alpha=1\,|\,\vec{s}^{(m)})}_{\langle\tau_\alpha\rangle_{|\vec{s}^{(m)}}} - \langle\tau_\alpha\rangle_{RBM}$$

$$= \langle\tau_\alpha\rangle_D - \langle\tau_\alpha\rangle_{RBM}$$

⚠ Now $\langle\tau_\alpha\rangle_D$ depends on the model. ⚠

$$\frac{\partial \mathcal{L}}{\partial w_{i\alpha}} = \frac{1}{M}\sum_m s_i^{(m)}\, p(\tau_\alpha=1\,|\,\vec{s}^{(m)}) - \langle s_i\tau_\alpha\rangle_{RBM}$$

$$= \underbrace{\langle s_i\tau_\alpha\rangle_D}_{\text{positive term}} - \underbrace{\langle s_i\tau_\alpha\rangle_{RBM}}_{\text{neg. term}}$$

Learning algo: For a mini-batch

1) compute $p(\tau_\alpha\,|\,\vec{s}^{(m)}) \longrightarrow s_i^{(m)}\, p(\tau_\alpha=1\,|\,\vec{s}^{m})$

i) compute $p(\tau_a | \vec{s}^{(m)}) \longrightarrow s_i^{(m)} p(\tau_a = 1 | \vec{s}^{m})$

also $\langle s_i \rangle_D$

ii) Sample N_s conf. for T timesteps starting from rdm init. cdt.

$$\longrightarrow \{ \vec{s}^{T,(m)}, \tau^{T,(m)} \}_{m=1,\dots,N_s}$$

$$\langle \tau_a \rangle_{RBM} \simeq \frac{1}{N_s} \sum_m p(\tau_a^T = 1 | \vec{s}^{T,(m)})$$

$$\langle s_i \rangle_{RBM} \simeq \frac{1}{N_s} \sum_m s_i^{T,(m)}$$

$$\langle s_i \tau_a \rangle_{RBM} \simeq \frac{1}{N_s} \sum_m s_i^{T,(m)} \tau_a^{T,(m)}$$

iii) Update your parameters:

$$\theta_i \longleftarrow \theta_i + \alpha \left(\langle s_i \rangle_D - \langle s_i \rangle_{RBM} \right)$$

α is the learning rate

$$\eta_a \longleftarrow \eta_a + \alpha \left(\langle \tau_a \rangle_D - \langle \tau_a \rangle_{RBM} \right)$$

$$w_{ia} \longleftarrow w_{ia} + \alpha \left(\langle s_i \tau_a \rangle_D - \langle s_i \tau_a \rangle_{RBM} \right)$$

III) Mean-Field Phase diagram

Can we understand what happen in the small couplings limit?

assume W_{ia} is small : $s_i \simeq \langle s_i \rangle + \delta_i$ $\langle s_i \rangle = \langle \tau_a \rangle = 1/2$

$\tau_a \simeq \langle \tau_a \rangle + \delta_a$

$$\underbrace{\langle \tau_a \rangle}_{1/2} + \delta_a = \mathrm{sig}\Big(\sum_i W_{ia} \big(\langle s_i \rangle + \delta_i \big) + \eta_a \Big)$$

(If $\eta_a = -\sum_i W_{ia} \langle s_i \rangle$)

$$= \frac{1}{1 + e^{-\sum_i W_{ia}\delta_i}} \simeq \frac{1}{1 + \big(1 - \sum_i W_{ia}\delta_i\big)}$$

$$\simeq \frac{1}{2\big(1 - \sum_i \frac{W_{ia}\delta_i}{2}\big)} \simeq \frac{1}{2}\Big[1 + \frac{1}{2}\sum_i W_{ia}\delta_i\Big]$$

$$\cancel{\frac{1}{2}} + \delta_a \simeq \cancel{\frac{1}{2}} + \frac{1}{4} W_{ia}\delta_i$$

$$\delta_a \simeq \frac{1}{4} W_{ia}\delta_i \rightarrow \vec{\delta}^{hid} = \frac{1}{4} W^T \cdot \vec{\delta}^{(vis)}$$

$$\vec{\delta}^{vis} = \frac{1}{4} W \cdot \vec{\delta}^{(hid)}$$

Those perturbation are amplified if the strongest eigenvalue of W :

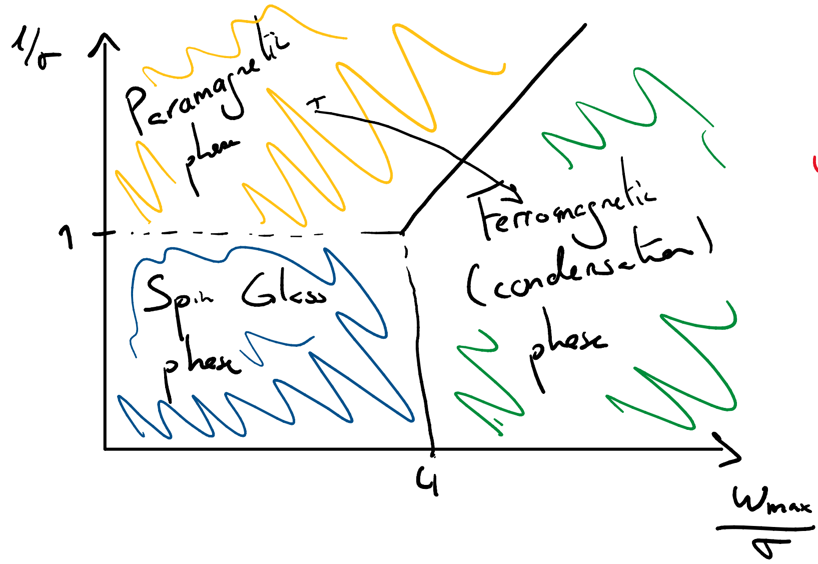
$$\boxed{w_{max} > 4}$$

otherwise they are killed

If we assume the following decomposition : $W_{ia} = \underbrace{\sum_{\alpha=1}^{K} u_i^\alpha w_\alpha v_a^\alpha}$ + r_{ia}

rdm gauss.

$1/\sigma$ $\uparrow$ 



a contribution of $k \sim O(1)$ isolated eigenvalues

rdm gauss. noise with variance σ

U & V are left and right eigenvectors

Singular Value Decomposition

Ferromagnetic phase: the system condenses over the strongest mode of W_{ia}

$$\begin{cases} m_\alpha = \dfrac{1}{\sqrt{N_v}} \sum_i \langle s_i \rangle u_i^\alpha \\ \bar{m}_\alpha = \dfrac{1}{\sqrt{N_h}} \sum_a \langle \tau_a \rangle v_a^\alpha \end{cases}$$

become $O(1)$

IV] Learning behavior

⚠ for this section I will consider $\begin{cases} s_i = \pm 1 \\ \tau_a = \pm 1 \end{cases}$; $\begin{matrix} \theta_i = 0 \\ \eta_a = 0 \end{matrix}$

We have seen that the SVD decomposition of W plays an important role.

$$\cdot \; W_{ia} = \sum_{\alpha=1}^{\min(N_v, N_h)} u_i^\alpha w_\alpha v_j^\alpha$$

$\rightarrow$ project the gradient over $\vec{u}^\alpha$ and $\vec{v}^\alpha$

$$\left(\frac{\partial \mathcal{L}}{\cdot}\right) = \sum u_i^\alpha \frac{\partial \mathcal{L}}{\cdot} v_a^\beta = \sum u_i^\alpha \left[\langle s_i \tau_a \rangle - \langle s_i \tau_a \rangle \right] v_a^\beta$$

$$\left(\frac{\partial \mathcal{L}}{\partial \underline{w}}\right)_{\alpha\beta} := \sum_{i,c} u_i^\alpha \frac{\partial \mathcal{L}}{\partial w_{ic}} v_c^\beta = \sum_{i,c} u_i^\alpha \left[\langle s_i \tau_c\rangle_D - \langle s_i \tau_c\rangle_{RBM}\right] v_c^\beta$$

$$\begin{bmatrix} s_\alpha := \sum_i s_i u_i^\alpha \\ \tau_\alpha := \sum_c \tau_c v_c^\alpha \end{bmatrix} = \langle s_\alpha \tau_\alpha\rangle_D - \langle s_\alpha\tau_\alpha\rangle_{RBM} \qquad (*)$$

Gradient upd

$$\underline{w}(t+\delta t) \approx \underline{w}(t) + \delta t \boxed{\frac{d\underline{w}}{dt}} = \frac{\partial\mathcal{L}}{\partial \underline{w}}$$

$$\left(\frac{d\underline{w}}{dt}\right)_{\alpha\beta} = \sum_{i,c} u_i^\alpha \left(\frac{d}{dt} w_{ic}\right) v_c^\beta = \sum_{i,c} u_i^\alpha \left(\frac{d}{dt}\sum_\gamma u_i^\gamma w_\gamma v_c^\gamma\right) v_c^\beta$$

$$= \sum_{i,c,\gamma}\Big[\underbrace{\underbrace{u_i^\alpha u_i^\gamma}_{\sum\to\delta_{\alpha\gamma}} \frac{dw_\gamma}{dt} \underbrace{v_c^\gamma v_c^\beta}_{\downarrow\ \delta_{\gamma\beta}}}_{\delta_{\alpha\beta}\frac{dw_\alpha}{dt}} + \underbrace{u_i^\alpha \frac{du_i^\gamma}{dt} w_\gamma \underbrace{v_c^\gamma v_c^\beta}_{\delta_{\gamma\beta}}}_{\vec u^\alpha \frac{d\vec u^\beta}{dt} w_\alpha (1-\delta_{\alpha\beta})} + \underbrace{u_i^\alpha u_i^\gamma w_\gamma \frac{dv_c^\gamma}{dt} \cdot v_c^\beta}_{(1-\delta_{\alpha\beta}) w_\beta \frac{d\vec v^\alpha}{dt}\cdot\vec v^\beta}\Big]$$

$$= \delta_{\alpha\beta}\frac{dw_\alpha}{dt} + (1-\delta_{\alpha\beta})\left[\vec u^\alpha\cdot\frac{d\vec u^\beta}{dt} w_\alpha + w_\beta \vec v^\beta\cdot\frac{d\vec v^\alpha}{dt}\right] \qquad (**)$$

using (*) & (**)

$$\frac{dw_\alpha}{dt} = \langle s_\alpha\tau_\alpha\rangle_D - \langle s_\alpha\tau_\alpha\rangle_{RBM}$$

$$\underbrace{\vec u^\alpha \frac{du^\beta}{dt}}_{\to\ \text{infinitesimal}} w_\alpha = \mathcal{J}\left(\langle s_\alpha \tau_\beta\rangle_{D,\ \alpha\neq\beta}\,;\ \langle s_\alpha\tau_\beta\rangle_{RBM,\ \alpha\neq\beta}\right) \to \text{rotation of the orthog. matrix } u$$

$d \neq p$ $\qquad$ $d \neq p$ $\qquad$ the orthg. matrix. u

→ infinitesimal rotations operator over the vec. u → the same for. V.

→ when $w_{i\alpha}$ is small

$$\frac{\partial \mathcal{L}}{\partial w_{i\alpha}} = \frac{1}{M} \sum_m s_i^{(m)} \, \mathrm{th}\left(\sum_j w_{j\alpha} s_j^{(m)} \right) - \langle s_i \tau_\alpha \rangle_{RBM}$$

$$\simeq \frac{1}{M} \sum_m s_i^{(m)} \sum_j w_{j\alpha} s_j^{(m)} - w_{i\alpha}$$

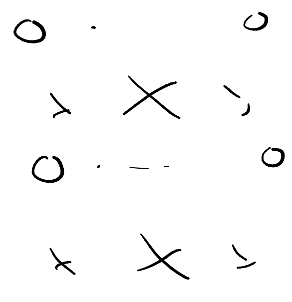
$$\simeq \sum_j C_{ij} \cdot w_{j\alpha} - w_{i\alpha} \qquad \text{when } C_{ij} := \frac{1}{M} \sum_m s_i^{(m)} s_j^{(m)}$$

$$\frac{d w_\alpha}{dt} \simeq w_\alpha \left[\langle s_\alpha^2 \rangle_D - 1 \right]$$

→ ~ transition behavior very similar to the one we get in GMM

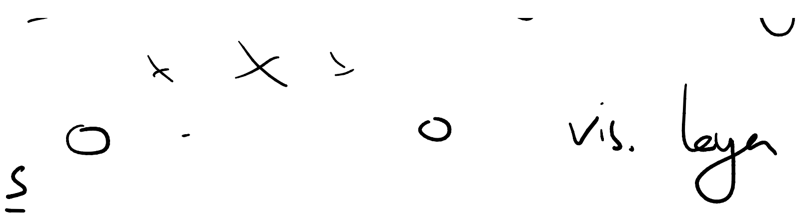
C_{ij} projected on the U matrix

V] Deep RBM ?



$\underline{\sigma}$: second hid lay

$\underline{\tau}$: first hid. lay.



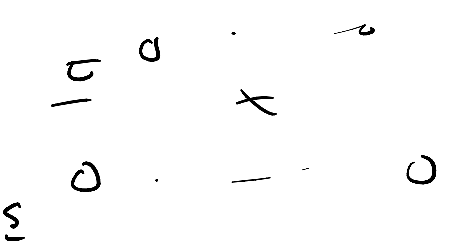
$$\mathcal{H} = -\sum_{i,k} s_i W^{(1)}_{ik} \tau_k - \sum_i \theta_i s_i - \sum_k \eta_k \tau_k - \sum_{k,\mu} \tau_k W^{(1)}_{k\mu} \sigma_\mu - \sum_\mu \delta_\mu \sigma_\mu$$

→ you still have conditional dependence:

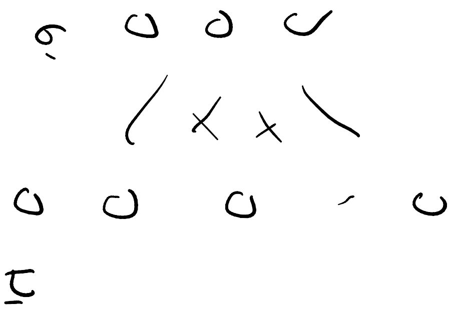
- if you fix $\underline{\tau}$ → $\{\underline{s}, \underline{\sigma}\}$ become indep
- —— $\boxed{\underline{s} \not\perp \underline{\sigma}}$ → $\underline{\tau}$ ——

⇝ Simplification: <u>Deep Belief Network</u>

They consist in stacking RBM:



RBM$_1$

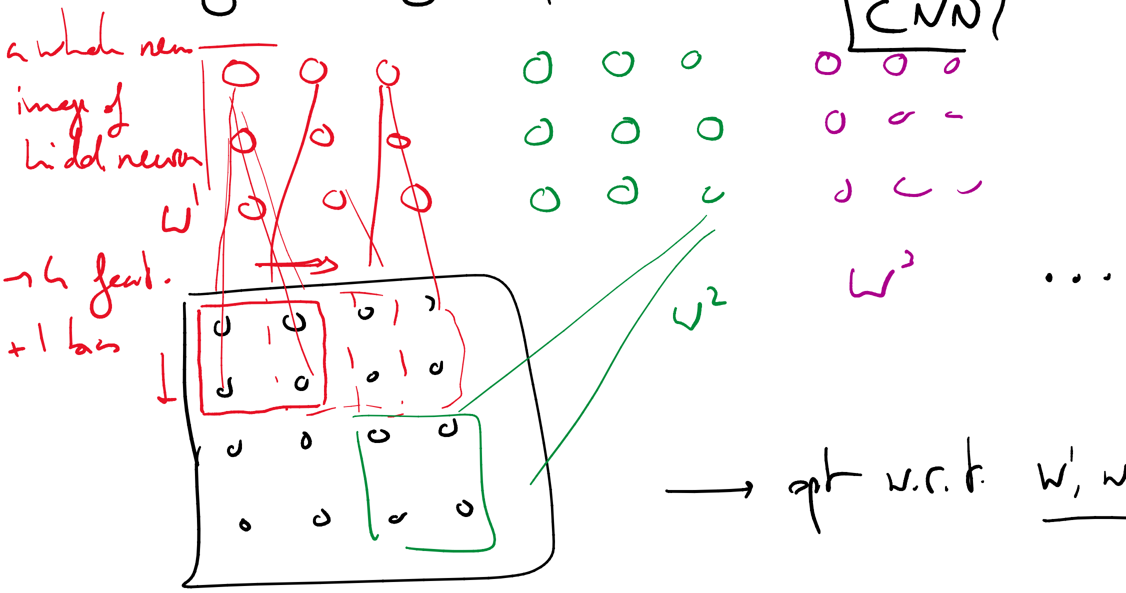


RBM$_2$

D] Few words on Conv. neural Network

& Generative adversarial Network

1) One of the very important layers: convolutional NN $\boxed{CNN}$

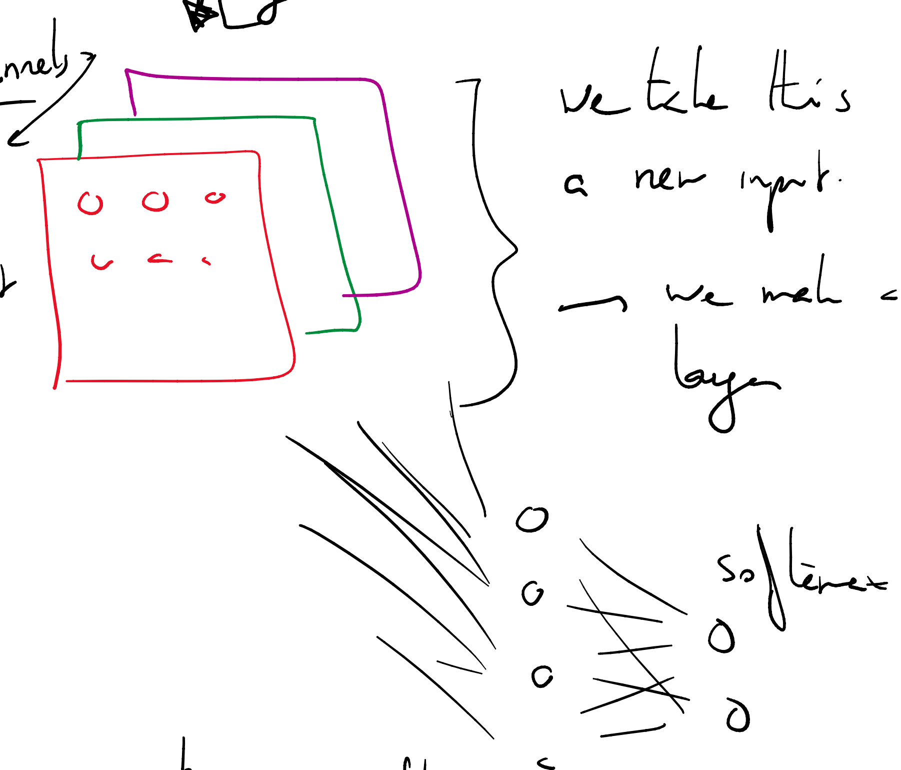


→ opt w.r.t. $W^1, W^2, W^3, \dots$

With images: invariant by translation

3 channels

after this: this the output of your conv. layer



we take this as a new input.

→ we make a perceptron layer

Im → conv → perceptron → softmax

Im → conv layer → perceptron (Dense) → softmax

Thx to CNN → impressive results on image classification

II) GAN: generative model without sampling

Good Fellow

You design 2 neural network

1 is the generator: from a set of ~ N_I rdm input (e.g. Gaussian iid.)

it transforms it using a nn: $G(\vec{z})$

$\vec{z} \in \mathbb{R}^{N_I}$

$G(\vec{z}) \in \mathbb{R}^{N_x \times N_y}$

2 the $\boxed{\text{discriminator}}$ D it takes an image $N_x \times N_y$

→ it outputs the proba that this image belong to the dataset or not

Training: 2 pass

1) you fix the weights of D

you opt G such that it "forces" D to answer that the samples from G are in the dataset

ii) you fix G

you opt D to answer correctly

E] Approximation to compute the negative terms

a] Normal Gibbs sampling

$\vec{s}^{(0)} \sim \text{unif}\{0,1\}$ → MC sampling starting from $\vec{s}^0$ up to $\{\vec{s}^T\}$

$$\langle s_i \tau_a \rangle_{RBM} \simeq \frac{1}{N_s} \sum_{m=1}^{N_s} s_i^{T,(m)} \, p(\tau_a = 1 | \vec{s}^{T,(m)})$$

I] Hinton : Constrative Divergence (CD-k)

$\{\vec{S}^{(0)}\}$ = data used in the minibatch

$$\langle s_i \tau_k \rangle_{RBM} \simeq \frac{1}{M} \sum_m \vec{S}^{(k),(m)} \cdot p(\tau_k = 1 \mid \vec{S}^{(k),(m)})$$

$\vec{S}^{(k)}$: vis after k MC steps

II] Persistent CD -k

for the first update : $\vec{S}^{0} \sim$ unif $\{0,1\}$

→ make k MC. steps

$$\langle s_i \tau_k \rangle_{RBM} \simeq \frac{1}{N_s} \sum_m s_i^{(m)} \, p(\tau_k = 1 \mid \vec{S}^{(k),(m)})$$

Iter: for the successive updates

→ you take the previous final state of the MC chain as initial cond for $\vec{S}^{(0)}$