Backpropagation: $\sigma(z)=\sigma(z)(1-\sigma(z))$

$\frac{\partial x_i^\ell}{\partial x_j^{\ell-1}} = \sigma'^{\ell}(\langle w_i^\ell, x^{\ell-1}\rangle) w_{ij} = J_{ij}^\ell \Rightarrow J^\ell = (J_{ij}^\ell)$

Many layers: $\frac{\partial x_i^\ell}{\partial x_j^{\ell-n}} = \sum_m \underbrace{\frac{\partial x_i^\ell}{\partial x_m^{\ell-1}}}_{J_{im}^\ell} \cdot \frac{\partial x_m^{\ell-1}}{\partial x_j^{\ell-n}} \Rightarrow \frac{\partial x^\ell}{\partial x^{\ell-n}} = J^\ell \cdot J^{\ell-1} \cdots J^{\ell-n+1}$

From layer ℓ to output: $\nabla_{x^\ell}^T \mathcal{L} = \nabla_y^T \mathcal{L} \cdot J^L \cdot J^{L-1} \cdots J^{\ell+1}$

Weights: $\frac{\partial \mathcal{L}}{\partial w_{ij}^\ell} = \frac{\partial \mathcal{L}}{\partial x_i^\ell} \cdot \frac{\partial x_i^\ell}{\partial w_{ij}^\ell} = \frac{\partial \mathcal{L}}{\partial x_i^\ell} \cdot \sigma'^{\ell}(\langle w_i^\ell, x^{\ell-1}\rangle) x_j^{\ell-1}$

Haar Wavelets: Example 4D: $\phi(x)=\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$ Mother wavelet $W(x)=\begin{pmatrix}1\\1\\-1\\-1\end{pmatrix}$, $W(2x)=\begin{pmatrix}1\\-1\\0\\0\end{pmatrix}$ $W(2x-1)=\begin{pmatrix}0\\0\\1\\-1\end{pmatrix}$

U: $d\times d$, always normalize!

All wavelets orthogonal!

$W_{n,k}(x) = 2^{\frac{n}{2}} W(2^n x - k)$ $\forall n,k$ integers, $n \geq 0$, $W_{n,k}$ nonzero on $[k\cdot 2^{-n}, (k+1)2^{-n}]$

$\int_{\mathbb{R}} W_{n,k}(x)\,dx = 0$, $\|W_{n,k}(x)\|^2_{L^2(\mathbb{R})} = \int_{\mathbb{R}} W_{n,k}^2(x)\,dx = 1$

Pairwise orthogonal: $\int_{\mathbb{R}} W_{n_1,k_1}(x)\cdot W_{n_2,k_2}(x)\,dx = \delta_{n_1,n_2}\cdot\delta_{k_1,k_2}$

Coefficient $X_w(k,n) = \int x \cdot W_{n,k}(x)\,dx = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} W\left(\frac{x-b}{a}\right) x\,dx$ $\begin{cases}1 & \text{if } n_1=n_2\\ 0 & \text{else}\end{cases}$, a=scaling, b=time

$X(a,b)$

Generative CNN: x=image, x_i=pixel, 2 extremes:

- all pixels independent: $p(x)=p(x_1)\cdot p(x_2)\cdots$
$\Rightarrow$ sample each pixel independently
- $p(x)=\prod_{i=1}^{n^2} p(x_i|x_1,\dots,x_{i-1})$ cond. on all previous pixels

Approach to learn a Generative Model: Pixel CNN:
condition only on previous & generated pixels:
convolutional mask to learn $p(x)$ by shifting over training images (generated / not yet)

Matching Pursuit (MP): Approximation of x onto U using K entries, r=residual

Objective: $z^* = \underset{z}{\arg\min} \|x - Uz\|_2$ s.t. $\|z\|_0 \leq K$: 1. Init $z \leftarrow 0$, $r \leftarrow x$ 2. while $\|z\|_0 < K$

3. select atom with smallest angle: $i = \underset{j}{\arg\max} |\langle u_j, r\rangle|$, 4. $z_i \leftarrow z_i + \langle u_i, r\rangle$, 5. $r \leftarrow r - \langle u_i, r\rangle u_i$

U orthogonal! If $\langle u_i, u_i^\wedge\rangle \neq 1 \Rightarrow r^{(n)}$ no longer orthogonal to $u_i^\wedge$

Exact recovery, when $K < \frac{1}{2}\left(1 + \frac{1}{m(U)}\right)$

Compressive Sensing: Why gathering so much data, when most of it will be removed (compressed) again?

Gather directly compressed! $x \in \mathbb{R}^D$, $x = Uz$ s.t. $\|z\|_0 = K$ $\hat{=}$ K-sparse on U^q. M measurements $y \in \mathbb{R}^M$

$y_i = \langle w_i, x\rangle \hat{=} M$ lin. comb. of x. $y = Wx = WUz = \Theta z$, Θ: $M\times D$, $M<D$. Reconstruction: more unknowns than equations: Find x from y: find $z^* = \underset{z}{\arg\min}\|z\|_0$ s.t. $y = \Theta z$ (e.g. with MP) or $\|z\|_1$: convex (z = overcomplete repr. of y)

Given z, reconstruct $x = Uz$

Sufficient conditions for reconstruction:

1. $W_{ij} \sim \mathcal{N}(0, \frac{1}{D})$, 2. $M \geq cK\ln\left(\frac{D}{K}\right)$ c some constant

Dual Ascent (in general, not RPCA):

Recover Lagrangian: $\min f(x)$ s.t. $Ax=b \Rightarrow \mathcal{L} = f(x) + \lambda^T(Ax-b)$

Dual Ascent = Gradient method for Dual problem:

(Dual = argmax!): $\lambda^{t+1} = \lambda^t + \eta^t \nabla D(\lambda^t)$

$\nabla D(\lambda^t) = Ax^* - b$, for $x^* = \underset{x}{\arg\min}\,\mathcal{L}(x, \lambda^t)$

(Dual: $\max \mathcal{L}(x^*, \lambda)$ s.t. $x^* = \arg\min \mathcal{L}(x, \lambda)$)

Dual Decomposition: f seperable $f(x) = f_1(x_1) + \dots + f_N(x_N)$ and $Ax = \sum_i A_i x_i = b$ also seperable, A_i = matrix!

$\Rightarrow$ also $\mathcal{L}$ seperable: $\mathcal{L} = \left(\sum_i \mathcal{L}_i\right) - \lambda^T b$, $\mathcal{L}_i = f_i(x_i) + \lambda^T A_i x_i$

Dual Decomposition for Dual Ascent:

$x_i^{t+1} = \underset{x_i}{\arg\min}\,\mathcal{L}_i(x_i, \lambda^t)$, do $\forall i$ and then update λ:

$\lambda^{t+1} = \lambda^t + \eta^t\left(\sum_{i=1}^N A_i x_i^{t+1} - b\right)$

Dual Ascent converges only under strict convex $f(x)$

$\Rightarrow$ overcome that: ADMM

$A = \begin{pmatrix}a & b & c\\ d & e & f\\ g & h & i\end{pmatrix}$ $\det(A) = aei + bfg + cdh - gec - hfa - idb$

Sparse Coding:

Dictionary U: $d\times d$, $\mathcal{D}$, x, z: $d\times 1$

Coding: $z = U^T x$, thresholding $z \to \hat{z}$: $\hat{z}_d = \begin{cases}0 & \text{if } |z_d| < \epsilon\\ z_d & \text{else}\end{cases}$

Reconstruction: $\hat{x} = U\hat{z}$ Pro: efficient coding: $z = U^T x$. Con: only sparse for specific signal

Error: $\|x - \hat{x}\|^2 = \sum_{d\notin\sigma} \|\langle x, u_d\rangle \cdot u_d\|^2 = \sum_{d\notin\sigma} \langle x, u_d\rangle^2$

σ = set of nonzero z_d, $x = \sum_{d=1}^D z_d(x)\cdot u_d$, $z_d(x) = \langle x, u_d\rangle$

$\hat{x} = \sum_{d\in\sigma} z_d(x)\cdot u_d$ $\hat{x}$ often better repr. than x bcs denoised

- Thresholding $\hat{=}$ same as PCA, but in PCA, dictionary is data dependent, here, dictionary apriori fixed.
- SVD $\hat{=}$ dictionary = rank 1 matrices
- In PCA, we get maximum of principal components
$\Rightarrow$ cannot learn overcomplete dictionary.

Dictionary Learning: Adapt dictionary to signal. Pro: achieve same approximation error as overcomplete but with smaller L = # atoms

Con: have to solve a matrix factorization problem: X: $D\times N$, U: $D\times L$, Z: $L\times N$

$(U^*, Z^*) = \underset{U,Z}{\arg\min}\|X - UZ\|_F^2$ s.t. Z = sparse, U = normalized, non-convex

convex in U or Z only. Iter Greedy Minimization:

Initialization: U^0 chosen: A) $u_\ell^0 \sim \mathcal{N}(0, I_D) \Rightarrow u_\ell^0 = \frac{u_\ell^0}{\|u_\ell^0\|_2}$ normalize $\hat{=}$ sampled from unitsphere

B) sample from X: $n \sim$ Uniform$(1,N) \Rightarrow u_\ell^0 = \frac{x_n}{\|x_n\|_2}$, C) Use fixed overcomplete dictionary (DCT, Haar wavelet, FFT)

1. Coding step: $Z^{t+1} = \underset{Z}{\arg\min}\|X - U^t Z\|_F^2$ s.t. Z sparse $\Rightarrow$ seperable into N problems for each column one: $z_n^{t+1} = \underset{z}{\arg\min}\|z\|_0$ s.t. $\|x_n - U^t z\|_2 \leq \sigma\cdot\|x_n\|_2$ where σ = chosen design parameter. 2. Dict update step: $U^{t+1} = \underset{U}{\arg\min}\|X - UZ^{t+1}\|_F^2$

s.t. $\|u_\ell\|_2 = 1\ \forall \ell \in [L]$: set $U = [u_1^t, \dots, u_\ell, \dots, u_L^t]$: $\min_{u_\ell}\|X - UZ^{t+1}\|_F^2 =$

$= \min_{u_\ell}\|\underbrace{X - \sum_{m\neq\ell} u_m^t (z_m^{t+1})^T}_{R_\ell^t} - u_\ell (z_\ell^{t+1})^T\|_F^2 = \min_{u_\ell}\|R_\ell^t - u_\ell (z_\ell^{t+1})^T\|_F^2$ s.t. $\|u_\ell\|_2 = 1$

Solution: $R_\ell^t \overset{\text{SVD}}{=} \tilde{U}\Sigma\tilde{V}^T = \sum_i \sigma_i \tilde{u}_i \tilde{v}_i^T \Rightarrow u_\ell^* = \tilde{u}_1$ ($\|u_\ell^*\|_2 = 1$ naturally satisfied) solution not perfect, only approx.

Also update $z_\ell = \tilde{v}_1$ but only temporary until all u updated, then use actual z vectors for real z update

D = dimension, L = # atoms in dictionary. $L > D \Rightarrow$ over-complete

$\Rightarrow$ more unknowns than equation in reconstruction $\Rightarrow$ no closed form solution $\Rightarrow$ add sparseness constraint: $z^* = \arg\min\|z\|_0$ s.t. $x = Uz$ $\hat{=}$ perfect reconstruction (exch. obj. and constraint for k-sparse approximation of reconstruction)

np-hard $\Rightarrow \|z\|_0 \to \|z\|_1$: convex, or stay with $\|z\|_0$ and us Matching Pursuit = efficient approximation

$\frac{L}{D}$ = over completeness factor Pro: sparse coding for several signal classes, Con: optimization

Coherence $m(\cdot)$ = measure of linear dependency

The higher $m(U)$, the worse for the reconstruction

$m(U) = \max_{i,j:\, i\neq j} |u_i^T u_j|$. $U\mathcal{O} \Rightarrow m(U) = 0$, Adding atom u to orthogonal $B \Rightarrow m([B, u]) \geq \frac{1}{\sqrt{D}}$

Assumed additive Gaussian noise $x = Uz + n$, $n \sim \mathcal{N}(0, \sigma^2 I)$

$\Rightarrow z^* = \underset{z}{\arg\min}\|z\|_0$ s.t. $\|x - Uz\|_2^2 < D\cdot\sigma^2$

or $z^* = \underset{z}{\arg\min}\|x - Uz\|_2$ s.t. $\|z\|_0 \leq K$, K given

or use $n = Uy$: $\hat{x} = Uz + Uy = U(z+y)$: noise cannot be sparse encoded $\Rightarrow y$ has a lot small coefficients $\Rightarrow$ threshold

Fast Fourier Transform (FFT): global support, good for "sine like" signal, bad for local signal. $f(t) = \frac{4}{\pi}\sum_{k=1}^{\infty}\frac{1}{n}\sin\left(\frac{k\cdot\pi\cdot t}{20}\right)$, orthogonal dict.

Gabor wavelets: directional oscillation, amplitude modulated by Gaussian window

$g(n_1, n_2; \mu_1, \mu_2, f, \theta) \propto \exp[-(n_1 - \mu_1)^2]\exp[-(n_2 - \mu_2)^2] \times \cos(f\cdot(n_1\cos\theta + n_2\sin\theta))$

Robust PCA: Assume $X = L_0 + S_0$ (L_0, S_0 true values $\hat{=}$ perfect): Approximate $X = L + S$

L = low rank = signal, S is sparse = noise. $\min_{L,S}$ rank$(L) + \mu\|S\|_0$ s.t. $L + S = X$.

Non convex $\Rightarrow$ change to $\min_{L,S}\|L\|_* + \mu\|S\|_1 \hat{=}$ Principal Component Pursuit (PCP) (nuclear)

(in general not the same but convex relaxation useful)

Perfect reconstruction is not possible if: L is low-rank, or S is sparse, or X is low rank and sparse (i.e. $X = \begin{pmatrix}0 & 1\\ 0 & 0\end{pmatrix}$).

Coherence conditions: L_0 cannot be sparse: $L_0 = U\Sigma V^T$, r = rank(L_0), ν = const > 0

L_0: $n\times n$: $\|U^T e_i\|^2 \leq \frac{\nu r}{n}$, $\|V^T e_i\|^2 \leq \frac{\nu r}{n}$, $|UV^T|_{ij}^2 \leq \frac{\nu r}{n^2} \Rightarrow$ principal components must not be sparse

Theorem: if L_0 satisfies coherence cond. for some $\nu > 0$, and

L_0: rank$(L_0) \leq \rho_r n \nu^{-1}(\ln(n))^{-2}$ $\rho_r > 0$; S_0: $n\times n$: $\|S_0\|_0 = m \leq \rho_s n^2$ $\rho_s > 0$

then: with prob. $1 - O(n^{-10})$, PCP with $\mu = \frac{1}{\sqrt{n}}$ recovers true solution $L^* = L_0$, $S^* = S_0$

same holds for rectangular matrices and $\mu = \frac{1}{\sqrt{\max(n,m)}}$

e_i = unit vector $\begin{pmatrix}0\\ \vdots\\ 1\end{pmatrix}$ i-th dim

~~Alternating Direction~~ Method of Multipliers:

Augmented Lagrangian $\mathcal{L}_\rho(x, \lambda) = f(x) + \lambda^T(Ax - b) + \frac{\rho}{2}\|Ax - b\|_2^2$

For any feasible x: $\mathcal{L}_\rho(x, \lambda) = \mathcal{L}(x, \lambda) = f(x)$

Method of Multipliers:

$x^{t+1} = \underset{x}{\arg\min}\,\mathcal{L}_\rho(x, \lambda^t)$, $\lambda^{t+1} = \lambda^t + \rho(Ax^{t+1} - b)$

$\Rightarrow$ additional penalty for violating constraints

$\Rightarrow$ converges under more general assumptions than Dual Ascent

Why ρ as const. step size?: Optimality conditions: "Dual feasibility":

$\nabla_x \mathcal{L}(x, \lambda) = \nabla_x f(x) + A^T\lambda \overset{!}{=} 0$

1. order optimality of x^{t+1}:

$0 = \nabla_x \mathcal{L}_\rho(x^{t+1}, \lambda^t) = \nabla_x f(x^{t+1}) + A^T(\lambda^t + \rho(Ax^{t+1} - b))$
$= \nabla f(x^{t+1}) + A^T\lambda^{t+1}$

$\Rightarrow$ using ρ as step size: iterate (x^{t+1}, λ^{t+1}) always dual feasible!

Not completely separable (and parallelizable) any more, but split x into 2 directions x_1, x_2: still seperable, but not parallelizable:

Alternating Direction Method of Multipliers (ADMM):

$\min_{x_1, x_2} f_1(x_1) + f_2(x_2)$ s.t. $A_1x_1 + A_2x_2 = b$, f_1, f_2 convex

Augmented Lagrangian:

$\mathcal{L}_\rho(x_1, x_2, \lambda) = f_1(x_1) + f_2(x_2) + \lambda^T(A_1x_1 + A_2x_2 - b) + \frac{\rho}{2}\|A_1x_1 + A_2x_2 - b\|_2^2$

ADMM: $x_1^{t+1} = \underset{x_1}{\arg\min}\,\mathcal{L}_\rho(x_1, x_2^t, \lambda^t)$,

$x_2^{t+1} = \underset{x_2}{\arg\min}\,\mathcal{L}_\rho(x_1^{t+1}, x_2, \lambda^t)$

$\lambda^{t+1} = \lambda^t + \rho(A_1x_1^{t+1} + A_2x_2^{t+1} - b)$

ADMM for RPCA: $f_1(L) = \|L\|_*$, $f_2(S) = \mu\|S\|_1$

$A_1x_1 + A_2x_2 = b \Rightarrow L + S = X$

($\min \|L\|_* + \mu\|S\|_1$ s.t. $L + S = X$)

$\min f(x) + f(x)$ $\Rightarrow \min f(x_1) + f(x_2)$ s.t. $x_1 - x_2 = 0$ ($x_1 = x_2$) $\Rightarrow$ ADMM

Augmented Lagrangian: vec$(\cdot)$ = concatenate all matrix inputs into giant vector; mat$(\cdot)$ = convert vector to matrix

$\mathcal{L}_\rho(L, S, \lambda) = \|L\|_* + \mu\|S\|_1 + \langle\lambda, \text{vec}(L + S - X)\rangle + \frac{\rho}{2}\|L + S - X\|_F^2$

$L^{t+1} = \underset{L}{\arg\min}\,\mathcal{L}_\rho(L, S^t, \lambda^t) = \mathcal{D}_{\rho^{-1}}(X - S^t - \rho^{-1}\text{mat}(\lambda^t))$

$S^{t+1} = \underset{S}{\arg\min}\,\mathcal{L}_\rho(L^{t+1}, S, \lambda^t) = S_{\mu\rho^{-1}}(X - L^{t+1} - \rho^{-1}\text{mat}(\lambda^t))$

$\lambda^{t+1} = \lambda^t + \rho\cdot\text{vec}(L^{t+1} + S^{t+1} - X)$ where $S_\tau(x) = \text{sgn}(x)\max(|x| - \tau, 0)$ elementwise

$\mathcal{D}_\tau(X) = U S_\tau(\Sigma) V^T$ where $X = U\Sigma V^T$ (SVD)

Collaborative Filtering: obj. stays same, constraints change to $L_{ij} + S_{ij} = X_{ij}\ \forall (i,j) \in J$ observed, bcs missing values

Matrix Calculus: $\frac{\partial}{\partial x}(b^Tx)=\frac{\partial}{\partial x}(x^Tb)=b$, $\frac{\partial}{\partial x}(x^Tx)=2x$, $\frac{\partial}{\partial x}(x^TAx)=(A^T+A)x$
$\frac{\partial}{\partial x}(Ax)=A$, $\frac{\partial}{\partial x}(x^TA)=A^T$, $\frac{\partial}{\partial x}(c^TAx)=A^Tc$, $\frac{\partial}{\partial X}(c^TXb)=cb^T$, $\frac{\partial}{\partial X}(c^TX^Tb)=bc^T$
$\frac{\partial}{\partial x}(\|x-b\|_2)=\frac{x-b}{\|x-b\|_2}$, $\frac{\partial}{\partial x}(\|x\|_2^2)=\frac{\partial}{\partial x}(\|x^Tx\|_2)=2x$, $\frac{\partial}{\partial X}(\|X\|_F^2)=2X$

Eigenvalue and Eigenvector: Eigenvalue problem: $Ax=\lambda x$
1. $\det(A-\lambda I)=0\Rightarrow\lambda_i$; 2. $\forall\lambda_i: (A-\lambda_iI)x_i=0\Rightarrow x_i$ = eigenvector
3. normalize $u_i=\frac{u_i}{\|u_i\|_2}$ $\lambda=0$ is normal eigenvalue. 2 same eigenvalues ⇒ 2 eigenvectors for it.

Eigendecomposition: $A\in\mathbb{R}^{N\times N}$ then $A=U\Lambda U^{-1}$ $U: N\times N$
if all λ nonzero: $A^{-1}=U\Lambda^{-1}U^{-1}$ with $(\Lambda^{-1})_{ii}=\frac{1}{\lambda_i}$, Asymm: $A=U\Lambda U^T$ ($U\diamond$)
Power method for finding λ_1, u_1: $v_{t+1}=\frac{Av_t}{\|Av_t\|}\xrightarrow{t\to\infty}u_1$, $\lambda_1=\lim_{t\to\infty}\frac{\|Av_t\|}{\|v_t\|}$ ⇒ A diagonalizable

Norms: ℓ_0: $\|x\|_0=|\{i|x_i\neq0\}|$
Nuclear: $\|X\|_*=\sum_{i=1}^{s}\sigma_i$ $s=\min(m,n)$, Euclidean $\|x\|_2=\sqrt{\sum_{i=1}^N x_i^2}=\sqrt{x^Tx}$
p-Norm: $\|x\|_p=\left(\sum_{i=1}^N|x_i|^p\right)^{\frac1p}$, $\|M\|_1=\sum_{i,j}|m_{ij}|$, $\|M\|_2=\sigma_{max}$, $\|M\|_p=\max_{v\neq0}\frac{\|Mv\|_p}{\|v\|_p}$
Frobenius: $\|A\|_F=\sqrt{\sum_{i,j}|A_{ij}|^2}=\sqrt{trace(AA^T)}=\sqrt{trace(A^TA)}=\sqrt{\sum_{i=1}^{\min(m,n) \text{ or } rank(A)}\sigma_i^2}$
$trace(XYZ)=trace(ZXY)$, $trace(AB)=trace(BA)$, $U\diamond$: $\|A\|_2=\|UA\|_2$

PCA: X: D×N Objective: $u=\arg\min_{\|u\|=1}\frac1N\sum_{i=1}^N\|\langle u,x_i\rangle-x_i\|_2^2$
$\tilde x_i-x_i=(I-uu^T)x_i=r_i\Rightarrow\frac1N\sum r_ir_i^T=\Sigma-\lambda uu^T\Rightarrow\arg\max_{\|u\|=1}\frac1N\sum\langle u,x_i\rangle^2$
$=\arg\max_{\|u\|=1}u^T\Sigma u\Rightarrow\mathcal L=u^T\Sigma u+\lambda(1-\langle u,u\rangle)$ $\frac{\partial\mathcal L}{\partial u}=0\Rightarrow u$ = ev of Σ
$\mathcal L$ = max for $u=\lambda_1$ 1. $\bar x=\frac1N\sum x_i$, $M=[\bar x,...,\bar x]$, 2. $\tilde X=X-M$ (destroys sparseness)
3. Covariance $\Sigma=\frac1N\sum(x_i-\bar x)(x_i-\bar x)^T=\frac1N\tilde X\tilde X^T$ 4. Eigenvalue decomp. $\Sigma=U\Lambda U^T$ (Σ symm.) Sort eigenvalues and vectors $\lambda_1\geq\lambda_2\geq...\geq\lambda_D$
5. K<D first K eigenvalues ⇒ U_K, Λ_K 6. $\bar Z=U_K^T\tilde X$, $\tilde{\bar X}=U_K\bar Z=U_KU_K^T\tilde X$
7. $\bar X=\tilde{\bar X}+M$, For compression, save $U_K,\bar Z,\bar x$ $U_K=[u_1,...,u_K]$: D×K
U_K: D×K, Σ: D×D, $\bar Z$: K×N, $\bar X$: D×N $=...=\sum_{i=K+1}^D\lambda_i\cdot trace(u_iu_i^T)$
error: $\frac1N\sum_{i=1}^N\|\tilde x_i-\bar x_i\|_2^2=\frac1N\|\tilde X-\bar X\|_F^2=\frac1N\|(U_KU_K^T-I_d)\tilde X\|_F^2$

SVD: A: M×N, U: M×M, D: M×N, V: N×N $A=UDV^T=\sum_{k=1}^r\sigma_ku_kv_k^T$ r=rank(A)
r=#σ>0 $\sigma_k=(D)_{kk}\geq0$ $A^T=VD^TU^T$, U,V$\diamond$: U^TU, V^TV, UU^T, VV^T all = I
$D=diag(\sigma_1,...,\sigma_s)$ s=min(m,n) U columns = left sing. vectors, V col = right
#distinct σi ≤ s, σi with 2 (or more) linear ind. sing. vectors = degenerate
$(D^+)_{ij}=\frac{1}{D_{ij}}$, D^+: N×M (transpose), kernel(A) = {x s.t. Ax=0} = nullspace = null(A)
range(A) = {z | ∃x s.t. Ax=z} dim(kernel(A)) + dim(range(A)) = N (rank(A))
Columns of U with σi>0 span the range of A, Asymm: SVD=PCA!
Columns of V with σi=0 span the null space of A ⇒ those columns can be any vector orthogonal to rest in V
delete zero columns of D and corresponding columns in V (= rows in V^T)
Reduced SVD: n>m (otherwise transpose):
U: m×m, D: m×m, V^T: m×n
Compute SVD:
1. sing. val = square root of eigenvalues of A^TA 2. Eig vec of A^TA = col of V
3. Eigvectors of AA^T = columns of U (or $U=AVD^{-1}$) 4. normalize col of U,V

Eckart-Young theorem: Use k largest σi with their singular vectors:
$A_k=\sum_{i=1}^k\sigma_iu_iv_i^T=U_kD_kV_k^T$ minimizes Frobenius and Euclidean error subject to constraint $rank(A_k)=k$. Each $\sigma_iu_iv_i^T$ = rank 1 matrix
k=rank(A) ⇒ perfect reconstruction
Error Frobenius: $\|A-A_k\|_F^2=\sum_{i=k+1}^{rank(A)}\sigma_i^2$, Euclidean: $\|A-A_k\|_2=\sigma_{k+1}$
$trace(A^TA)=trace(VD^TDV^T)=trace(D^TD)=trace(diag(\sigma_1^2,...,\sigma_r^2))=\sum_i^r\sigma_i^2$

Collaborative Filtering: Eckart-Young with SVD bad, bcs missing values need to be filled first (i.e. with average value) Better I = {(i,j) observed}
$\min_{rank(B)=k}\underbrace{\sum_{(i,j)\in I}(a_{ij}-b_{ij})^2}_{f(B)}$ ⇒ incomplete matrix ⇒ SVD not possible - hard to optimize
⇒ reparametrize $B=U^TV$ U: k×m, V: k×n ⇒ rank(B) ≤ k due to parametrization
$b_{ij}=\langle u_i,v_j\rangle\Rightarrow f(U,V)=\|A-U^TV\|_F^2\cdot\frac{1}{|I|}$ Alternating Minimization
$U^*\leftarrow\arg\min_U f(U,V)\Rightarrow V^*\leftarrow\arg\min_V f(U^*,V)$ ⇒ repeat
$f(U,V)=\frac{1}{|I|}\sum_j\sum_{(i,j)\in I}(a_{ij}-\langle u_i,v_j\rangle)^2$: U fixed: j separate least squares min a problems
Regularized: $f(u,v)+\mu(\|U\|_F^2+...)$ similar for V fixed ⇒ SGD: sample ij ⇒ optimize

Orthogonal matrices:
Permutation matrix: $A(\pi)=a_{ij}=\begin{cases}1 & \text{if } \pi_i=j\\0 & \text{else}\end{cases}$
Reflector matrix with direction u (‖u‖=1)
$A=I-2uu^T$ = projection on hyperplane $(\mathbb{R}u)^\perp$
$diag(\sigma_1,...,-\sigma_i,...,\sigma_n)\cdot(I-2e_ie_i^T)=diag(\sigma_1,...,\sigma_i,...,\sigma_n)$, Rotation: $A=\begin{pmatrix}\cos\alpha&-\sin\alpha\\\sin\alpha&\cos\alpha\end{pmatrix}$

Optimization: general assumption: f(x) continuous and differentiable.
Convex sets: $x,y\in Q\Rightarrow\alpha x+(1-\alpha)y\in Q\ \forall\alpha\in[0,1]$
- A,B conv ⇒ A∩B conv. (not A∪B)
- set of rank=k matrices is not convex
- projection on convex set is unique
- convex comb: $x=\lambda_1x_1+...+\lambda_nx_n$ $\sum\lambda_i=1$, $\lambda_i\geq0$

Convex function: domain of f (set of input values) convex and $f(\alpha x+(1-\alpha)y)\leq\alpha f(x)+(1-\alpha)f(y)$ $\forall\alpha\in[0,1]$ Examples: linear a^Tx, affine a^Tx+b, exponential exp(ax), every norm, $\max(x_1,...,x_n)$
Non convex: f(u,v)=u·v
epigraph = {(x,t) | x ∈ dom f, f(x) ≤ t}
convex opt. if f(x) & Q convex!

Coordinate descent: only update d^{th} coord (or set of some coords)
$d\sim[1,D]$: $u^*=\arg\min_u f(x_1^t,...,x_{d-1}^t,u,...,x_D^t)$
$x_d^{t+1}=u^*$, $x_i^{t+1}=x_i^t$ i≠d Repeat

Gradient descent = Steepest descent
gradient $\nabla f(x)=\begin{pmatrix}\frac{\partial f}{\partial x_1}\\\vdots\\\frac{\partial f}{\partial x_m}\end{pmatrix}$ $x^{t+1}=x^t-\eta\nabla f(x^t)$
Theorem: $f(x^t)-f(x^*_{opt})\leq\frac{c}{t}$ ⇒ step size $\eta\propto\frac1t>0$
subgradient g of f(x), if:
$f(y)\geq f(x)+g^T(y-x)$
Theorem $\leq\frac{c}{\sqrt t}$ for subgradient

Stochastic gradient descent (for additive objectives $f(x)=\frac1n\sum_i f_i(x)$)
Sample i uniformly $x^{(t+1)}=x^{(t)}-\eta\nabla f_i(x^{(t)})$
$\eta\propto\frac1t$ Full gradient often intractable for large datasets. r uniformly
$\mathbb{E}[\nabla f_i(x)]=\sum_i P(r=i)\nabla f_i(x)=\sum_i\frac1n\nabla f_i(x)=\nabla f(x)$

Convex relaxation:
- via projected gradient descent: solve for convex set, project on unconstrained
- rank(M) → $\|M\|_*$
- $\|M\|_0$ → $\|M\|_1$

Projected Gradient descent:
Solve unconstrained optimization and then project on constraint set
Projection: $P_Q(x)=\arg\min_{\tilde x\in Q}\|\tilde x-x\|$
$x^{(t+1)}=P_Q[x^{(t)}-\eta\nabla f(x^{(t)})]$
$x^{(t+1)}$ is unique if Q convex

KKT conditions: 1) $\nabla\mathcal L(x^*,\lambda^*,\nu^*)=0$ 2) $g_i(x^*)\leq0$, $h_j(x^*)=0$, $\lambda_i^*\geq0$ 3) $\lambda_i^*g_i(x^*)=0$ $f(x^*)=\mathcal L(x^*,\lambda^*,\nu^*)$

Lagrangian Multipliers:
min f(x) s.t. $g_i(x)\leq0$, $h_j(x)=a_j^Tx-b_j=0$
Lagrangian $\mathcal L(x,\lambda,\nu)=f(x)+\sum_i\lambda_ig_i(x)+\sum_j\nu_jh_j(x)$
$\lambda_i\geq0$, $\nu_j\in\mathbb R$ Dual fct: $d(\lambda,\nu)=\inf_x\mathcal L(x,\lambda,\nu)$ (min)
$d(\lambda,\nu)\leq f(x^*)$ Dual problem $\max_{\lambda,\nu}d(\lambda,\nu)$ s.t. $\lambda\geq0$
if dom f and f convex: $d(\lambda^*,\nu^*)=f(x^*)$
recover x^* from $\mathcal L()$: $x^*=\arg\min_x\mathcal L(x,\lambda^*,\nu^*)$
unconstrained now

Non-Negative Matrix Factorization: (not eigenfaces but parts of faces)
Topic Models: pLSA = probabilistic Latent Semantic Analysis
Better for word occurences than SVD bcs in SVD, can be <0 ⇒ no sense. d=document, z=topic=latent variable, w=word
$p(w|d)=\sum_{z=1}^K p(w,z|d)=\sum_{z=1}^K p(w|z)p(z|d)$ where $p(w|d,z)\overset{\text{cond. ind}}{=}p(w|z)$
$p(w,d)=p(w|d)p(d)$ or $p(w,d)=\sum_{z=1}^K p(w|z)p(d|z)p(z)$
p(w|d) = prob of w occurring in d, p(w|d) matrix of model
Sum up data in co-occurrence matrix X: N×D
x_{ij}: # occurrences of word w_j in document d_i
Alternatively: $\mathcal X$ = set over index pairs (i,j)
Log-likelihood: $\ell(U,V)=\sum_{i,j}x_{ij}\ln\sum_z\underbrace{p(w_j|z)}_{v_{zj}}\underbrace{p(z|d_i)}_{u_{zi}}=\sum_{(i,j)\in\mathcal X}\ln\sum_z v_{zj}\cdot u_{zi}$
$u_{zi},v_{zj}\geq0$ can be neglected bcs implicitly considered in KKT cond. $\sum_z u_{zi}=1\ \forall i$, $\sum_j v_{zj}=1\ \forall z$
Introduce variational parameters q_{zij} and use Jensen's ineq.
$\ln\sum_z q_{zij}\frac{u_{zi}v_{zj}}{q_{zij}}\geq\sum_z q_{zij}(\ln u_{zi}+\ln v_{zj}-\ln q_{zij})=a\cdot b$, $a=\sum_{ij}x_{ij}\cdot b$
$\sum_z q_{zij}=1$, $q_{zij}\geq0$ $\mathcal L()=a+\underbrace{\lambda(\sum_z q_{zij}-1)}_{\text{not a constraint but true}}+\sum_i\alpha_i(\sum_z u_{zi}-1)+\sum_z\beta_z(\sum_j v_{zj}-1)$
max $\mathcal L()$ unconstrained ⇒ Expectation-Maximization (EM)
E-Step: $\frac{\partial\mathcal L()}{\partial q_{zij}}=0\Rightarrow q_{zij}^*=\frac{u_{zi}\cdot v_{zj}}{\sum_k u_{ki}\cdot v_{kj}}=p(z|d_i,w_j)$ = post. distr. prob. that (d_i,w_j) comes from z
M-Step: $\frac{\partial\mathcal L()}{\partial u_{zi}}=0$, $\frac{\partial\mathcal L()}{\partial v_{zj}}=0\Rightarrow u_{zi}^*=\frac{\sum_j x_{ij}q_{zij}}{\sum_j x_{ij}}$, $v_{zj}^*=\frac{\sum_i x_{ij}q_{zij}}{\sum_{i,l}x_{il}\cdot q_{zil}}$
Convergence guaranteed, maybe local opt!

Generative document model = sample new row of X = new document
Latent Dirichlet Allocation (LDA) = Bayesian version of pLSA
u_i = prob. vector: $p(u_i|\alpha)\propto\prod_{z=1}^K u_{zi}^{\alpha_z-1}$ V = real parameters, U = latent ⇒ averaged out
$\ell=\sum_j x_j$ = document length $\pi_j=\sum_z v_{zj}u_z$
$p(x|V,u)=\frac{\ell!}{\prod_j x_j!}\cdot\prod_j\pi_j^{x_j}\Rightarrow p(x|V,\alpha)=\int p(x|V,u)p(u|\alpha)du$
Model:
- For each d_i, sample $u_i\sim$ Dirichlet(α)
- For each word slot w^t $1\leq t\leq\ell$: - sample topic $z^t\sim Multi(u_i)$ - sample word $w^t\sim Multi(v_{z^t})$

NMF Algorithm for quadratic cost (ALS) = Alternating Least squares:
X: N×M, $x_{ij}\geq0$: $X\approx U^TV$, $x_{ij}=\sum_z u_{zi}\cdot v_{zj}=\langle u_i,v_j\rangle$
cost $J(U,V)=\frac12\sum_{i,j}(x_{ij}-\langle u_i,v_j\rangle)^2=\frac12\|X-U^TV\|_F^2$
s.t. $u_{zi},v_{zj}\geq0\ \forall i,j,z$ if low-rank approx: also s.t. $rank(U^TV)=k$
if regularized: cost $J(U,V)+\|U\|_F+\|V\|_F$
Algorithm: Init U,V=rand()
Update U: solve normal equation $(VV^T)U=VX^T$ numerically with gradient descent methods, project $u_{zi}=\max(0,u_{zi})$. Update V same with normal equation $(UU^T)V=UX$ and project also, SGD
$\frac{\partial J(U,V)}{\partial V}=(-U)(X-U^TV)=0^T$ single column of V: $(UU^T)v_j=Ux_j$

Word embeddings: $w=w^1,...,w^T$ sequence of words V = whole vocabulary
$I=\{-R,...,-1,1,...,R\}$ window of offsets. Skip-gram: $p_\theta(w|w')$ = prob that w occurs in context window of w'
log-likelihood = predictive score:
$\ell(\theta,w)=\sum_{t=1}^T\sum_{\Delta\in I}\ln p_\theta(w^{t+\Delta}|w^t)$. Represent words as latent vectors (=embeddings) $w\to(x_w,b_w)\in\mathbb R^{d+1}$ (bias)
Log-bilinear: $\ln p_\theta(w|w')=\langle x_w,x_{w'}\rangle+b_w+c$
b_w big for frequent words: $b_w\uparrow\Rightarrow p_\theta(w|w')\uparrow\ \forall w'$ ⇒ high prob no matter what context
Exponentiate ⇒ soft-max: $p_\theta(w|w')=\frac{\exp(\langle x_w,x_{w'}\rangle+b_w)}{\sum_{v\in V}\exp(\langle x_v,x_{w'}\rangle+b_v)}$

angle$(x_w,x_{w'})$ small ⇒ $\langle x_w,x_{w'}\rangle\uparrow=p(w|w')\uparrow$ ⇒ $x_w,x_{w'}$ similar.
Model parameters $\theta=((x_w,b_w)_{w\in V})\in\mathbb R^{(d+1)\cdot|V|}$
Complete log-likelihood:
$\sum_{t=1}^T\sum_{\Delta\in I}[\underbrace{b_{w^{(t+\Delta)}}+\langle x_{w^{(t+\Delta)}},x_{w^{(t)}}\rangle}_{\#1}-\underbrace{\ln\sum_{v\in V}\exp(\langle x_v,x_{w^{(t)}}\rangle+b_v)}_{\#2}]$ #2 intractable for V≈1 million
Modification #1: Context vectors: Distinguish main voc V and Context Voc C: different embeddings $x_w\in V$, $y_w\in C$ ∀words
$\ln p_\theta(w|w')=\langle y_w,x_{w'}\rangle+b_w$. Pro: model flexibility, Con: dimensionality
Modification #2: make model tractable: Negative sampling:
Modify likelihood to: $G(z)=\frac{1}{1+e^{-z}}$ = logistic sigmoid
$\ell(\theta,w)=\sum_{t=1}^T\sum_{\Delta\in I}\ln G(f(w^{t+\Delta},w^t))+k\mathbb E_v[\ln G(-f(v,w^t))]$
$f(v,w)=\langle y_v,x_w\rangle+b_v$ k = # negative samples
Pro: tractable, con: need to define sample distr. for negative words (i.e. frequency of each word one possibility)
Alternative Modification #2: Global Vectors = GloVe (non-convex): Co-occurrence matrix $N=(n_{ij})\in\mathbb R^{|V|\cdot|C|}$
n_{ij} = # occurences of $w_i\in V$ in context of $w_j\in C$
N is sparse and computable in one pass over the text
Weighted least squares with unnormalized $\tilde p_\theta(w_i|w_j)=\exp[\langle x_i,y_j\rangle+b_i+c_j]$, Weighting fct. $f(n)=\min(1,(\frac{n}{n_{max}})^\alpha)$
$\alpha\in[0,1]$, max = chosen parameter to limit influence of frequent words. max ≠ |N|!
$\tilde p_\theta(w_i|w_j)$ should neither be too large nor too small, bcs unnormalized!
least-squares obj: $H(\theta,N)=\sum_{i,j}f(n_{ij})(\ln\underbrace{n_{ij}}_{target}-\ln\underbrace{\tilde p_\theta(w_i|w_j)}_{model})^2$
Full gradient often intractable ⇒ SGD: sample (i,j) s.t. $n_{ij}>0$
1. $x_i^{new}\leftarrow x_i+2\eta f(n_{ij})(\ln n_{ij}-\langle x_i,y_j\rangle)y_j$
2. $y_j^{new}\leftarrow y_j+2\eta f(n_{ij})(\ln n_{ij}-\langle x_i,y_j\rangle)x_i$
Matrix Decomposition: $m_{ij}=\log n_{ij}$ and absorb bias in vectors x,y: $x_{w,d+1}=1$, $x_{w,d}=b_w$, $y_{w,d+1}=c_w$, $y_{w,d}=1$
$f(\cdot)=1\ \forall i,j\Rightarrow\min H(\theta,N)=\min\|M-X^TY\|_F^2\Rightarrow$ SVD
$f(\cdot)=\begin{cases}1 & \text{if } n_{ij}>0\\0 & \text{else}\end{cases}\Rightarrow\min_{X,Y}\sum_{i,j:n_{ij}>0}(m_{ij}-(X^TY)_{ij})^2\Rightarrow$ SGD
Using normalized model $p_\theta(w)=\frac{\tilde p_\theta(w_i|w_j)}{\sum_{i\in V}\tilde p_\theta(w_i|w_j)}$ need to compute partition fct.
But then can use log-likelihood instead of least squares: $L=\sum_t\ln p_\theta(w_t)$ and problem of $\tilde p_\theta(w)$ being too large or too small is counterbalanced by normalization ⇒ the higher the better

K-means Algorithm (ALS)
Cost: $J(U,Z)=\|X-UZ\|_F^2=\sum_{i=1}^N\sum_{j=1}^K z_{ij}\|x_i-u_j\|_2^2$ (d×n, d×k, k×n) k=#clusters
$z_{ij}\in\{0,1\}$ = hard assignments $\sum_{j=1}^K z_{ij}=1\ \forall i$
Initialize centroids. E-Step: $z_{ij}^{(t)}=\begin{cases}1 & \text{if } j=\arg\min_s\|x_i-u_s^{(t-1)}\|_2\\0 & \text{else}\end{cases}$
M-Step: $\nabla_{u_j}J(U,Z)=0=-2\sum_i z_{ij}(x_i-u_j)\Rightarrow u_j^{(t)}=\frac{\sum_i z_{ij}^{(t)}x_i}{\sum_i z_{ij}}$ if $\sum_i z_{ij}\geq1$
guaranteed to converge to local optimum
K-means = special case of GMM:
- spherical shapes ⇒ $\Sigma_j=\sigma_j^2I$
- hard assignments σ→0

Gaussian Mixture Model (GMM)
Isotropic multivariate Gaussian: $\Sigma=\sigma^2I\Rightarrow\prod_{i=1}^D\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x_i-\mu_i)^2}{2\sigma^2}\right)$
else: $\frac{1}{(2\pi)^{\frac D2}|\Sigma|^{\frac12}}\exp\left(-\frac12(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$
Model: $p(x,\theta)=\sum_{j=1}^K\pi_j\mathcal N(x|\mu_j,\Sigma_j)$. Assignment variable $z\in\{0,1\}^K$, $\sum_{j=1}^K z_j=1$
Categorical distr. $Pr(z_j=1)=\pi_j$ or $p(z)=\prod_{j=1}^K\pi_j^{z_j}$, $\pi_j\geq0$ $\sum\pi_j=1$
$p(x;\theta_j)=\mathcal N(x;\mu_j,\Sigma_j)$

Everything here only for one data point x! Inference: given x, infer z
Joint distr. $p(x,z,\theta)=\prod_j(\pi_jp(x,\theta_j))^{z_j}$ = likelihood for one data point
Post. prob: $Pr(z_j=1|x)=\frac{Pr(z_j=1)p(x|z_j=1)}{\sum_l Pr(z_l=1)p(x|z_l=1)}=\frac{\pi_jp(x,\theta_j)}{\sum_l\pi_lp(x,\theta_l)}$
MLE (log-likelihood): $\arg\max_\theta\sum_{i=1}^N\ln\left(\sum_{j=1}^K\pi_jp(x_i,\theta_j)\right)=\hat\theta$
No closed form solution because ln(Σ...) ⇒ Jensens ineq.
$\ln p(x,\theta)=\ln\sum_j q_j\frac{\pi_jp(x,\theta_j)}{q_j}\geq\sum_j q_j(\ln p(x,\theta_j)+\ln\pi_j-\ln q_j)$
can be done for each data point (additive) π_k,μ_k,Σ_k,x_n all vectors or matrices
Expectation-Maximization (EM) for GMM:
1. Initialize $\pi_k^{(0)},\mu_k^{(0)},\Sigma_k^{(0)}$ for k=1,...,K
~~E-Step: decouple Lagrangian for each data point:~~
$\max_q\left\{\sum_{n=1}^N\sum_{k=1}^K q_{nk}[\ln p(x_n,\theta_k)+\ln\pi_k-\ln q_{nk}]+\sum_{n=1}^N\lambda_n\left(\sum_{k=1}^K q_{nk}-1\right)\right\}+\nu\left(\sum_{j=1}^K\pi_j-1\right)$
α = complete Lagrangian
E-Step: $\frac{\partial\alpha}{\partial q_{nk}}=\ln p(x_n,\theta_k)+\ln\pi_k-\ln q_{nk}-1+\lambda_n=0\Rightarrow q_{nk}=\frac{p(x_n,\theta_k)\pi_k}{e^{1-\lambda_n}}$
$\Rightarrow q_{nk}=\mathbb E[z_{nk}]=\frac{\pi_k^{(t-1)}\mathcal N(x_n|\mu_k^{(t-1)},\Sigma_k^{(t-1)})}{\sum_{j=1}^K\pi_j^{(t-1)}\mathcal N(x_n|\mu_j^{(t-1)},\Sigma_j^{(t-1)})}$
M-Step: Optimize expected complete log-likelihood wrt parameters:
$\mu_k^{(t)}=\frac{\sum_{n=1}^N q_{nk}x_n}{\sum_{n=1}^N q_{nk}}$, $\pi_k^{(t)}=\frac1N\sum_{n=1}^N q_{nk}$, $\Sigma_k^{(t)}=\frac{\sum_{n=1}^N q_{nk}(x_n-\mu_k)(x_n-\mu_k)^T}{\sum_{n=1}^N q_{nk}}$
Repeat. Guaranteed to converge to local opt.
k-means as initialization (bcs k-means is faster than GMM) ⇒ good initialization:
GMM means as k-means mean $\mu\to\mu_j$
$\Sigma_j=\left(\sum_{i=1}^{|M|}(x_i-\mu_j)(x_i-\mu_j)^T\right)\frac{1}{|M|}$ M = set of all points of cluster j
$\pi_j=\frac{\#\text{points in cluster } j}{N}=\frac{M}{N}$
Generative model: 1. sample cluster from categorical 2. given cluster, sample point from N()
Model selection: AIC/BIC for GMM: accuracy vs. complexity
smallest AIC/BIC = best number parameters. Single result meaningless ⇒ compare different #clusters
AIC/BIC
#clusters
Akaike Information Criterion (AIC)
$AIC(\theta|X)=-\ln(p(X;\theta))+\mathcal K(\theta)$
Bayesian Information Criterion (BIC)
$BIC(\theta|X)=-\ln(p(X;\theta))+\frac12\mathcal K(\theta)\ln(n)$
BIC penalizes complexity more than AIC
d=dimension, n=#points, k=#clusters
$\mathcal K(\theta)$=#parameters
Fixed covariance: $\mathcal K(\theta)=k\cdot d+(k-1)$
Full covariance: $\mathcal K(\theta)=k\left(d+\frac{d(d+1)}{2}\right)+(k-1)$

(Convolutional) Neural Networks:
Layer index l, neuron i: x^1 = raw input, $x^l=F^l(x^{l-1};W^l)=\sigma^l(W^lx^{l-1})$, $W^l=\begin{pmatrix}w_1^{lT}\\\vdots\\w_m^{lT}\end{pmatrix}$
Output: Regression: $y=W^Lx^{L-1}$, Binary classification (logistic) $y_1=p(Y=1|x)=\frac{1}{1+\exp(-\langle w_1^L,x^{L-1}\rangle)}$
Multiclass (softmax): $\hat y$ = NN output $\in[0,1]^K$ $y_k=P(Y=k|x)=\frac{\exp(\langle w_k^L,x^{L-1}\rangle)}{\sum_m\exp(\langle w_m^L,x^{L-1}\rangle)}$ Binary $w_1^L=w^L$ vector
Squared loss: $\ell(\hat y,y)=\frac12(\hat y-y)^2$ y = true label ∈ {0,1}
Cross-entropy loss: $\ell(\hat y,y)=-y\ln(\hat y)-(1-y)\ln(1-\hat y)$
Compare to logistic regression: $y=\frac{1}{1+\exp(-w^Tx)}$ ⇒ directly on input
NN: same function but on x^{L-1} = on learned representation / features
Empirical risk: $\mathcal L(\theta,X)=\frac1T\sum_{t=1}^T\ell(\hat y_t,y_t)$. Regularized: $\mathcal L(\cdot)+\frac\lambda2\|\theta\|_2^2$ or drop out
SGD for datapoint x_t: $\theta\leftarrow(1-\eta\lambda)\theta-\eta\nabla_\theta\ell(y_t,\hat y_t)$ zero padding and stride!
2D convolution: $F_{n,m}(x;w)=\sigma\left(b+\sum_{k=-2}^2\sum_{l=-2}^2 w_{k,l}\cdot x_{n+k,m+l}\right)$
Pooling = downsampling: average or max
CNN: conv-pool-conv-pool-...- full connected
CNN: highly non-convex