Week1

$MSE(f)=\frac{1}{N}\sum_{n=1}^N(y_n-f(x_n))^2$

$Convex: h(\lambda u+(1-\lambda)v)\le \lambda h(u)+(1-\lambda)h(v)$

Cvex f has a unique glob min w^*, sum of cvx = cvx

Convex set: $\mathcal{C}$ is convex iff $\theta u+(1-\theta)v\in\mathcal{C}$ Proof convexity: show that Hess is positive semi-def (all eigvals are ≥ 0)

(1) nonneg weighted sums (2) compo with affine mapping (3) poitwise max of convex funcs (4) restric to a line (5)

Week2

Gradient Descent $\mathcal{O}(ND)$

1. Start arbitrary $w_o\in\mathbb{R}$
2. For i do $w_{t+1}=w_t-\eta_t\nabla\mathcal{L}(w_t)$

MSE for linreg: $\mathcal{L}(w)=\frac{1}{2N}e^Te$

$\nabla\mathcal{L}(w)=-\frac{1}{N}X^Te$

Stochastic Gradient Descent (SGD) $\mathcal{O}(D)$

NB: Watch out that your loss has indeed this form before applying SGD!

$\mathcal{L}(w)=\frac{1}{N}\sum_{n=1}^N\mathcal{L}_n(w)$

$\mathcal{L}_n$ = cost at n-th training example (unbiased)

1. Start at an arbitrary $w_0\in\mathbb{R}^d$ 2. For $t=1,2,...$ do: Pick data point $(x',y')\in_{u.a.r.} D$

$w_{t+1}=w_t-\gamma\nabla\mathcal{L}_n(w_t;x',y')$

Mini-batch SGD

$g=\frac{1}{|B|}\sum_{n\in B}\nabla\mathcal{L}_n(w_t)$

$w_{t+1}=w_t-\gamma g$

Subgradient

$g\in\mathcal{R}^D$ is a subgradient iff: $\forall u. \mathcal{L}(u)\ge\mathcal{L}(w)+g^T(u-w)$

$w_{t+1}=w_t-\gamma g$ g = subgradient to $\mathcal{L}$ at w_t

NB: if $\mathcal{L}$ is convex and differentiable at w, the only subgrad at w is $g=\nabla\mathcal{L}(w)$. SubGD => $\mathcal{L}_n$ = subgrad

Projected Gradient Descent

add a projection ont convex set $\mathcal{C}$

$P_\mathcal{C}(w')=argmin_{v\in\mathcal{C}}||v-w'||$

$w_{t+1}=P_\mathcal{C}[w_t-\gamma\nabla\mathcal{L}(w_t)]$

Week3

Linear regression MSE

Normal equations: equations obt when setting the $\nabla=0$. $X^T(y-Xw)=0$ for lin reg

$\mathcal{L}(w)=\frac{1}{2N}||Xw-y||^2$

$\nabla\mathcal{L}(w)=0\Rightarrow w^*=(X^TX)^{-1}X^Ty$ We take $\hat y$ = projection of y onto $span(X)$. Prediction: $\hat y_m=x_m^Tw^*$

Gram Matrix X^TX only invertible if $rank(X)=D$

(1) if $D>N$ we have $rank(X)<D$ (we can still solve for rank-defficiency)(2) if $D\le N$ but some columns are collinear then X is ill-conditionned $Rightarrow$ issues

Closed form: $\mathcal{O}(D^2N)(X^TX)+\mathcal{O}(DN)(X^Ty)+\mathcal{O}(D^3)(inverseX^TX)=\mathcal{O}(D^2N)$

MLE for linear least-squares vs Gaussian

$p(y|X,w)\sim\mathcal{N}(w^Tx_i,\sigma^2)$:

$y_i=w^Tx_i+\epsilon_i,\epsilon_i\sim\mathcal{N}(0,\sigma^2)$

Maximizing the log likelihood:

$=\underset{w}{\text{argmin}}\sum_i^n(y_i-w^Tx_i)^2$ Gaussian noise on the data⇔ Least squares

Similarly for MAE and Laplace prior: $p(y|X,w)=\frac{1}{2b}e^{-\frac{1}{b}|y_n-x_n^Tw|}\Rightarrow min\frac{1}{b}\sum_{i=1}^n|w^Tx_i-y_i|$

Regularization $min_w\mathcal{L}(w)+\Omega w$

Ridge Regression

$min_w\frac{1}{2N}\sum_{n=1}^N[y_n-x_n^Tw]^2+\lambda||w||_2^2$

Gradient: $\nabla_wL(w)=-2\sum_{n=1}^N(y_n-w^Tx_n)x_n+2\lambda w=2[(X^TX+\lambda I)w-X^Ty];\mathcal{O}(D) for stoch$

$w^*_{ridge}=(X^TX+\lambda'I)^{-1}X^Ty,\lambda'=2N\lambda;\mathcal{O}(D^3+ND^2)$

eigvals($X^TX+\lambda'I$) $\ge\lambda'$ so the inverse always exists

Lasso: MSE and L1 regularization, encourages sparse w^*

Ridge reg = MAP estimate

Introduce bias by expressing assumption through a Bayesian prior $w_i\in\mathcal{N}(0,\beta^2)$

Bayes rule: $P(w|x,y)=\frac{P(w|x)P(y|x,w)}{P(y|x)}$

$=\underset{w}{\text{argmin}}\lambda||w||_2^2+\sum_{i=1}^n(y_i-w^Tx_i)^2$, $\lambda=\frac{\sigma^2}{\beta^2}$

Underfitting and Overfitting

Fight overfitting with (1)simpler model (2) more data

Week4

Generalization,Model Selection

True Risk: $L_\mathcal{D}(f)=\mathbb{E}_\mathcal{D}[\ell(y,f(x))]$, where loss fction $=\ell(y,f(x))=\frac{1}{2}(y-f(x))^2$ for ridge.

The True risk cannot be computed since $\mathcal{D}$ is not known.

Empirical Risk: $L_S(f)=\frac{1}{|S|}\sum_{(x_n,y_n)\in S}\ell(y_n,f(x_n))$ (for ridge, emp risk = MSE)

Training Risk: $L_S(f)=\frac{1}{|S|}\sum_{(x_n,y_n)\in S}\ell(y_n,f_S(x_n))$

$L_S(f_S)\ne L_D(f_S)$ because of overfitting.

Validation Risk: $L_{S_{test}}(f_{S_{train}})$ in expectation = True Risk

Generalization error: $G(f)=|L_\mathcal{D}(f)-L_{S_{test}}(f)|$

Gen error decreases as $\mathcal{O}(\frac{1}{\sqrt{|S_{test}|}})$

Model Selection

When testing K hpyer-parameters, the gen error goes up by a factor proportional to $\mathcal{O}(\sqrt{(ln(K))})$.

If we chose the "best"function according to the validation risk, then its true risk is not too far away from the true risk of the optimal choice. KFold CV: NBerror:

Bias-Variance Decomposition

For simple models f_S: the bias is large but $Var(L_D(f_S))$ is small.

$\mathbb{E}_{S_{train}\sim\mathcal{D},\epsilon\sim\mathcal{D}_\epsilon}[(f(x_0)+\epsilon-f_{S_{train}}(x_0))^2]=Var_{\epsilon\sim\mathcal{D}_\epsilon}[\epsilon]+(f(x_0)-\mathbb{E}_{S'_{train}\sim\mathcal{D}}[f_{S'_{train}}(x_0)])^2+\mathbb{E}_{S_{train}\sim\mathcal{D}}[(\mathbb{E}_{S'_{train}\sim\mathcal{D}}[f_{S'_{train}}(x_0)]-f_{S_{train}}(x_0))^2]$

Simpler: True Risk = Noise variance + $bias^2$ + variance Each of the term is a lower bound on the true error for any input x_0. The noise imposes a strict lower bound on what we can achieve.

Week5

Classification

(1) Estimating the probability (2) Quantize the value to a discrete value according to some threshold. Boundaries = decision boundaries.

Goal: fraction of misclassified cases is small: MSE ≠good match:counts + and - deviation eq bad, although only one of them can lead to a missclass.

KNN

$f_{S_{train},k}(x)=\frac{1}{k}\sum_{n:x_n\in nbh_{S_{train},k(x)}}y_n$ Large k = simple model, small k = complex model if N samples allow you to perform well in 1D you need $\mathcal{O}(N^D)$ samples in D dimensions.

Has a misclass rate at most 2x the one of the Bayes classifier if we have a lot of data.

MAP criterion for classification

We take the label decision which maximize the proba: $\hat y(x)=argmax_{y\in\mathcal{Y}}p(y|x)$.

$\mathbb{P}(\hat y(x)=y)=\int p(x)p(\hat y(x)|x)dx$

Logistic regression

we use $y_n\in+-1$ for svm and $y_n\in 0,1 for logistic$

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

we bet that $p(y|x,w)\sim Ber(y,\sigma(w^Tx))$

Link function: $\sigma(w^Tx)=\frac{1}{1+exp(-w^Tx)}$ (Sigmoid)

$logisticLoss[0-1](w^Tx)=log(1+e^{-yw^Tx})$

MLE for logistic regression

$argmax_wp(y,X|w)=argmax_wp(y|X,w)=\prod_{n=1}^N\sigma(x_n^Tw)^{y_n}[1-\sigma(x_n^Tw)]^{1-y_n}$

We can define:

$\mathcal{L}(w)=-\sum_{n=1}^Ny_nln\sigma(x_n^Tw)+(1-y_n)ln[1-\sigma(x_n^Tw)]=\sum_{n=1}^Nln[1+exp(x_n^Tw)]-y_nx_n^Tw$

so we chose: $w^*=argmin_w\mathcal{L}(w)$

$\nabla\mathcal{L}(w)=X^T[\sigma(Xw)-y]$

Newton Method for Logistic Regression

$Newtonupdate: w_{(t+1)}=w_t-\gamma_t(H_t)^{-1}\nabla\mathcal{L}(w_t)$

Regularized Logistic Regression

When data is linsep logreg tend to inf. $w^*=argmin_w-\sum_{n=1}^Nlnp(y_n|x_n^Tw)+\frac{\lambda}{2}||w||^2$ eq to MAP instd of MLE: $argmax_wp(w|X,y)=argmax_w\frac{p(X,y|w)p(w)}{p(X,y)}=argmax_wp(X,y|w)p(w)$

Still for classif take the most likely label.

Week6

Exponential Families

Expo family:: $p(y|\eta)=h(y)exp[\eta^T\phi(y)-A(\eta)]$

$\phi(y)$ is the sufficient statistic: given an independant set of samples and the empirical average of $\phi(y)$ we can optimally estimate η.

Link function: g relates the mean of $\phi(y)$ to the param η Constraint: $\int_yh(y)exp[\eta^T\phi(y)]dy=e^{A(\eta)}$.

$A(\eta)$ ensures a proper normalization (cumulant)

For every choice of $h(y)$, $\phi(y)$, η we will get a member of the expo family. Then chose $A(\mu)$ s.t. the expression is well normalized.

Lemma: $\nabla A(\eta)=\mathbb{E}[\phi(y)]$, $\nabla^2A(\eta)=\mathbb{E}[\phi(y)\phi(y)^T]-\mathbb{E}[\phi(y)]\mathbb{E}[\phi(y)]^T$

Poisson as expo: $p(y|\mu)=\frac{\mu^ye^{-\mu}}{y!}=\frac{1}{y!}e^{yln(\mu)-\mu}=h(y)e^{\eta\phi(y)-A(\eta)}$

So $\bullet h(y)=\frac{1}{y!}\bullet\phi(y)=y\bullet\eta=g(\mu)=ln(\mu)\bullet\mu=g^{-1}(\eta)=e^\eta$

Max Likelihood param estimation for expo

Goal: estimate the parameter η

$L(\eta)=-ln(p(y|\eta))=\sum_{n=1}^N[-ln(h(y_n))-\eta^T\phi(y_n)+A(\eta)]$ then $\eta=g(\frac{1}{N}\sum_{n=1}^N\phi(y_n))$

Generalized Linear Models

Given an element of the expo family with a scalar $\phi(y)$ we can construct from this a data model by assuming that a sample (x,y) follows:

Generalized linear model: $p(y|x,w)=h(y)e^{x^Tw\phi(y)-A(x^Tw)}$ We consider the cost function:

$L(w)=-\sum_{n=1}^Nlnp(y_n|x_n^Tw)=-\sum_{n=1}^Nln(h(y_n))+x_n^Tw\phi(y_n)-A(x_n^Tw)$

We use the fact that $\frac{dA(\eta)}{d\eta}=\mathbb{E}[\phi(y)]=g^{-1}(\eta)$ to obtain

$\nabla_w\mathcal{L}(w)=-\sum_{n=1}^Nx_n\phi(y_n)-x_ng^{-1}(x_n^Tw)$

matrix notation: $\nabla\mathcal{L}(w)=X^T[g^{-1}(Xw)-\phi(y)]=0$

Example for Logistic reg: $\nabla\mathcal{L}(w)=X^T[\sigma(Xw)-y]=0$

Curse of Dimensionality

Bayes Class: $f_*(x)=\mathbb{1}_{\{P(y=1|x)>1/2\}}$

Claim 1 : number of sample = $\mathcal{O}(N^D)$ the risk to stay constant.

Claim 2: As dimension increases the choice of KNN becomes random.

Lemma: $\mathbb{E}_{S_{train}}[\mathcal{L}(f_{S_{train}})]\le2\mathcal{L}(f_*)]+4c\sqrt{d}N^{-\frac{1}{d+1}}$

Week7

SVM

$w^*=argmin_w\sum_{n=1}^N[a-y_nx_n^Tw]_++\frac{\lambda}{2}||w||^2$

using Hinge loss: $[z]_+:=max\{0,z\}$

Duality

$\mathcal{L}(w)=max_\alpha G(w,\alpha)$ So since $[1-y_nx_n^Tw]_+=max_{\alpha_n\in[0,1]}\alpha_n(1-y_nx_n^Tw)$ then: $G(w,\alpha)$:

$min_wmax_{\alpha\in[0,1]^N}\sum_{n=1}^N\alpha_n(1-y_nx_n^Tw)+\frac{\lambda}{2}||w||^2=min_wmax_\alpha G(w,\alpha)$

In general: $max_xmin_yf(x,y)\le min_ymax_xf(x,y)$

Minimax Theorem: if $f(x,y)$ is convex in x and concave in y then

$max_ymin_xf(x,y)=min_xmax_yf(x,y)$

$Y=diag(y)$

$max_{\alpha\in[0,1]^N} \sum_{n=1}^N \alpha_n(1 - \frac{1}{\lambda} y_n x_n^T X^T Y\alpha) + \frac{\lambda}{2}\left\|\frac{1}{\lambda}X^TY\alpha\right\|$
= $max_{\alpha\in[0,1]^N}\alpha^T\mathbf{1} - \frac{1}{2\lambda}\alpha^T YXX^TY\alpha$
The dual is kenerlized with $K = XX^T$ and the solution α = nnz only for the training ex that are important for the decision boundary (inside margin,wrong side or on margin)

Coordinate descent
1. init $\alpha^{(0)}\in\mathcal{R}^N$
2. For t = 0:maxIter do: sample a rand coord n $\in$ 1..N
optimize g w.r.t n
$u^*\leftarrow armin_{u\in\mathcal{R}} g(\alpha_1^{(t)},..,u,..,\alpha_N^{(t)})$
update $\alpha_n^{(t+1)} = u^*, \alpha_{n'}^{(t+1)} = \alpha_{n'}^{(t)} \forall n\neq n'$

Kernel Trick
Solving Ridge: $\mathcal{O}(D^3+ND^2)$, via kernel: $w^*=X^T\alpha^*$ with $\alpha^*=(XX^T+\lambda I_N)^{-1}y$: $\mathcal{O}(N^3+DN^2)$
Representer Theorem: for all problem of the form $min_w \sum_{n=1}^N \mathcal{L}_n(x_n^Tw,y_n)+\frac{\lambda}{2}||w||^2$ there exist α^* s.t. $w^*=X^T\alpha^*$
Kernel Ridge: $\alpha^* = argmax_\alpha -\frac{1}{2}\alpha^T(XX^T+\lambda I_N)\alpha + \alpha^Ty$

Kernels
A kernel function is associated with a feature map:$\kappa(x,x')=\phi(x)^T\phi(x')$
Prove f=kernel: (1) find the associated feature map $\psi(x)$ show that $K(x,y)=\psi(x)^T\psi(y)$. (2) Verify Menger's therorem
Menger's Theorem:
1. K should be symmetric $\kappa(x,x')=\kappa(x',x)$
2. K should be positive semi-definite
NB: positive-definite functions give rise to ∞-dim feature space.
New prediction with kernel:
$\hat{y} = w^{*T}x_{test} = \sum_{n=1}^N k(x_{test},x_n)\alpha_n$ (here for id kernel) cost: $\mathcal{O}(N*kernel)$

Week8
k-mean
$\hat{R}(\mu)=\hat{R}(\mu_1,...,\mu_k)=\sum_{i=1}^n \min_{j\in\{1,...k\}} ||x_i-\mu_j||_2^2$
$\hat\mu = \text{argmin}_\mu \hat{R}(\mu)$
(Lloyd's heuristic):
Initialize cluster centers $\mu^{(0)}=[\mu_1^{(0)},...,\mu_k^{(0)}]$
While not converged
$z_i\leftarrow arg\min_{j\in\{1,...,k\}}||x_i-\mu_j^{(t-1)}||_2^2$; $\mu_j^{(t)}\leftarrow\frac{1}{n_j}\sum_{i:z_i=j}x_i$
NB: kmean enforces spherical clusters.

Gaussian Mixture Models
No spherical cluster:
$p(X|\mu,\Sigma,z)=\prod_{n=1}^N\prod_{k=1}^K[\mathcal{N}(x_n|\mu_k,\Sigma_k)]^{z_{nk}}$
No hard assignment: $p(z_n=k)=\pi_k$ where $\pi_k>0\forall k$ and $\sum_{k=1}^K\pi_k=1$

Gaussian Mixture Models in action
Prior: $p(z_n=k)=\pi_k$
Marginal likelihood(θ): $p(x_n|\theta)$
$p(x_n|\theta)=\sum_{k=1}^K p(x_n,z_n=k|\theta) = \sum_{k=1}^K p(z_n=k|\theta)p(x_n|z_n=k,\theta) = \sum_{k=1}^K\pi_k\mathcal{N}(x_n|\mu_k,\Sigma_k)$
Without latent variable: $\mathcal{O}(N)$ parameters, after marginalization:$\mathcal{O}(D^2K)$ (ass $D,K<<N$)

Week9
$Q(\theta,\theta^{(t)})=\mathbb{E}_{z|x,\tilde\theta}[logP(x,z|\theta)|x,\tilde\theta]$
we want $Q(\theta^{(t)},\theta^{(t-1)})\geq Q(\theta^{(t-1)},\theta^{(t-1)})$

EM algo
(1) E-step: For each k and n calculate $q_{kn}^{(t)}$
$q_{kn}^{(t)} = P(z_n = k|x_n,\theta^{(t-1)}) = \frac{\pi_k^{(t-1)}\mathcal{N}(x_n|\mu_k^{(t-1)},\Sigma_k^{(t-1)})}{\sum_{k=1}^K\pi_k^{(t-1)}\mathcal{N}(x_n|\mu_k^{(t-1)},\Sigma_k^{(t-1)})}$
(2) Compute marginal likelihood (cost): $\mathcal{L}(\theta_t)=\sum_{n=1}^N log\sum_{k=1}^K\pi_k^{(t-1)}\mathcal{N}(x_n|\mu_k^{(t-1)},\Sigma_k^{(t-1)})$
(3) M-step: Fit clusters to weighted data points (MLE marginal):
$\theta^{(t)}=argmax_\theta Q(\theta|\theta^{(t-1)})$
$\pi_k^{(t)}\leftarrow\frac{1}{n}\sum_{n=1}^n q_{kn}^{(t)}$; $\mu_k^{(t)}\leftarrow\frac{\sum_{n=1}^n q_{kn}^{(t)}x_n}{\sum_{n=1}^n q_{kn}^{(t)}}$
$\Sigma_k^{(t)}\leftarrow\frac{\sum_{n=1}^n q_{kn}^{(t)}(x_n-\mu_k^{(t)})(x_n-\mu_k^{(t)})^T}{\sum_{n=1}^n q_{kn}^{(t)}}$

SVD and PCA
SVD: $X=USV^T$ U,T unitary. Unitary matrices rotate, don't change norm.
Lemma: For any DxN matrix X and any DxN rank-K matrix $\hat{X}$:
$\left\|X-\hat X\right\|_F^2\geq\|X-U_KU_K^TX\|_F^2=\sum_{i\geq K+1}s_i^2$
where U_k is the D xK matrix consisting of the first K columns of U
We project on the first left singvecs of U.
We have:$U_KU_K^TX=U_KU_K^TUSV^T=US^{(K)}V^T$.

Week10
Matrix Factorization
$min_{W,Z}\mathcal{L}(W,Z)=\frac{1}{2|\Omega|}\sum_{(d,n)\in\Omega}[x_{dn}-(WZ^T)_{dn}]^2$: $W\in\mathcal{R}^{DxK}$, $Z\in\mathcal{R}^{NxK}$, $K<<D,N$
large K facilitates overfitting.
Regul:$min_{W,Z}\mathcal{L}(W,Z) = \frac{1}{2}\sum_{(d,u)\in\Omega}[x_{dn}-(WZ^T)_{dn}]^2+\frac{\lambda_w}{2}\|W\|_F^2+\frac{\lambda_z}{2}\|Z\|_F^2$
Identifiable: a model is identifiable iff $\theta_1\neq\theta_2\Rightarrow P_{\theta_1}\neq P_{\theta_2}$ i.e. 2 sets of different parameters imply 2 different proba distribs.

Alternating Least Squares
1. $Z^{*T}=(W^TW+\lambda_zI_K)^{-1}W^TX$ 2. $W^{*T}=(Z^TZ+\lambda_wI_K)^{-1}Z^TX^T$ cost as ridge regression per iteration. $\mathcal{O}(ND^2+D^3)$

Handling Text
Vocabulary: $\mathcal{V}=\{w_1..w_D\}$
Context: $\mathcal{C} = \{w_1...w_N\}$ (document, paragraph,window)
GloVe, (Global Vectors)unsup learn algo for obtaining vect repr for words;achieved by mapping words into space where the distance between w is prop to semantic similarity. train perf on co-occ. Combines matfact and local context window methods.
NB: we replace $x_{dn} := log(n_{dn})$ and obtain the count matrix X
$min_{W,Z}\mathcal{L}(W,Z) = \frac{1}{2}\sum_{(d,n)\in\Omega} f_{dn}[x_{dn}-(WZ^T)_{dn}]^2; f_{dn}=min\{1,(\frac{n_{dn}}{n_{max}})^\alpha\},\alpha\in[0,1]$
row(W) = rep(voc-w);row(Z)=rep(cont-w)
SG;Word2vec: uses logreg to sep real wp (w_d,w_n) from fake. Unsupervised.Streams through text,on the fly.Use neg-samplings;(NB: word2vec approximates glove)
textcolorpurple**Language models**: predict next word, multi-class(soft-max loss) Problem: you need to do the dot product with all words of the vocabulary.
FastText: Supervised sent classification:uses mat-fact: (sup and unsup aspects): Given sent$s_n = (w_1..w_m)$ let $x_n\in\mathcal{R}^V$ be its bow repr.
$min_{W,Z}\mathcal{L}(W,Z)=\sum_{s_n a sentence} f(y_nWZ^Tx_n)$
Z^Tx_n sums the words vectors (columns of Z) x to create a sentence rep: W classifies it. f =log loss function.

Week11
Neural Networks
Repr Lemma: for subsetof cont diff fctions, NN approx 1hidlayer, n nodes approx well in av any fction on bounded domain: $\int_{|x|\leq r}(f(x)-f_n(x))^2dx\leq\frac{(2Cr)^2}{n}$; $w_{i,j}^{(l)}$ = edge from node i in layer l-1 to node j in layer l; $z_j^{(l)}=\sum_i w_{i,j}^{(l)}x_i^{(l-1)}+b_j^{(l)}$; $\phi(x)=\frac{1}{1+e^{-x}}$; Rect in 3d: $g(x_1,x_2)=\phi(w(x_1+a_1))-\phi(w(x_1-b_1))+\phi(w(x_2+a_2))-\phi(w(x_2-b_2))\Rightarrow\phi(w(g(x_1,x_2)-3/2))$

Backpropagation
$\frac{\partial\mathcal{L}_n}{\partial w_{i,j}^{(l)}}=\sum_k\frac{\partial\mathcal{L}_n}{\partial z_k^{(l)}}\frac{\partial z_k^{(l)}}{\partial w_{i,j}^{(l)}}=\frac{\partial\mathcal{L}_n}{\partial z_j^{(l)}}\frac{\partial z_j^{(l)}}{\partial w_{i,j}^{(l)}}=\delta_j^{(l)}x_i^{(l-1)}$ Forward pass: $z^{(l)}=(W^{(l)})^Tx^{(l-1)}+b^{(l)}$,;$x^{(l)}=\phi(z^{(l)})$; $\delta_j^{(l)}=\frac{\partial\mathcal{L}_n}{\partial z_j^{(l)}}$ Bckward: Set $\delta^{(L+1)}=-2(y_n-x^{(L+1)})\phi'(z^{(L+1)})$; $\delta^{(l)}=(W(l+1)\delta^{(l+1)})*\phi'z^{(l)}$;
Final comp: $\frac{\partial\mathcal{L}_n}{\partial w_{i,j}^{(l)}}=\delta_j^{(l)}x_i^{(l-1)},\frac{\partial\mathcal{L}_n}{\partial b_j^{(l)}}=\delta_j^{(l)}$
Regularization $\Rightarrow$ weight decay.

Augmentation
Cropping, resizing, rotating, compressing (PCA), addition of noise

Dropout
$p_i^{(l)}$ = proba to "keep"the node i in layer l
Pred: (1) generate K subnets, average output (2) scale the output of node i at layer l by $p_i^{(l)}$.
Advantage: (1) limits overfitting (2) advantage of model averaging (bagging, ensemble methods)
CNN:consider input of size: $N1xN2$.run filter withoug padding, outputsize $(N_1+2(K_1-1))x(N_2+2(K_2-1))$ (valid padding)

Week12
White Box attack
Have access to the details of the algo i.e. we need access to the NN that implemented the prediction in order to compute the gradients.
$\tilde{x}=x-\epsilon h(x)\frac{\nabla_xg(x)}{\|\nabla_xg(x)\|_2}$
where $g(x)=\hat p(y|x), g(x)=$ ground truth.

Black Box attack
Cannot look inside the box.

Week13
Bayes Net
Goal: Recognize independence relationship given a Bayes Net. The graph needs to be acyclic to correspond to a valid factorization.
Lemma: X are cond indep of Y cond on Z if X and Y are D-separated by Z. (converse !nec true)
If X and Y are not D-separated, then there is distrib compat with the bayes net s.t. X and Y are cond dep given Z.
D-sep $\Leftrightarrow$ **every** path from X to Y is blocked by Z
Blocked: a path from X to Y is blocked by Z iff it contains a variable/node s.t either (1) a node $\in$ Z and is h2t or t2t (2)a node is h2h and neither this node nor any of its descendants are in Z.
NB: 2 variables are indep (given the empty set) if there is a h2h node between them.
NB: a h2h node blocks if none of its children or himself are in Z.
Markov Blanket: $X_j \perp\!\!\!\perp X_i|blanket(X_i)$

Factor graphs
$\sum_z f(z,..)=\sum_z\prod_{k=1}^K g_k(z,..)=\prod_{k=1}^K\sum_z g_k(z,..)$
NB:Bayes net is a tree $\Rightarrow$ F graph is a tree. converse not nec true.
NB: a cyclic graph cannot use the message passing algorithm!

Message passing Algorithm
Initialization:variable node $\mu(x)=1$, function node: $\mu(x)=f(x)$
f Node rule:$\mu(x)=\sum_{x_1...x_J}h(x,x_1,..,x_J)\prod_{j=1}^J\mu_j(x_j)$ x is the higher variable the factor is attached to
Variable Node rule: $\mu(x)=\prod_{k=1}^K\mu_k(x)$ all the μ_k depend on the variable x
Max complexity = $\mathcal{O}(|\mathcal{X}|^p), p=max_n deg(f_n)$