1. History

1.1 Perceptron

Threshold Unit
$f[w,b](x) = \operatorname{sign}(x \cdot w + b)$ where $x \cdot w := \sum_{i=1}^n x_i w_i$

Decision Boundary
$x \cdot w + b = 0 \Leftrightarrow \frac{x \cdot w}{\|w\|} + \frac{b}{\|w\|} = 0$

$x \cdot w - b = \begin{pmatrix} x \\ -1 \end{pmatrix} \cdot \begin{pmatrix} w \\ b \end{pmatrix} =: \tilde{x} \cdot \tilde{w}, \quad \tilde{x}, \tilde{w} \in \mathbb{R}^{n+1}$

Geometric Margin
$\gamma[w,b](x,y) := \frac{y(x \cdot w + b)}{\|w\|}$

Maximum Margin Classifier
$(w^*, b^*) \in \operatorname{argmax}_{w,b} \gamma[w,b](S)$
with $\gamma[w,b](S) := \min_{(x,y)\in S} \gamma[w,b](x,y)$

Perceptron Learning
if $f[w,b](x) \neq y$: update $w \leftarrow w + yx$, and $b \leftarrow b + y$
$w^0 \in \operatorname{span}(x_1, \ldots, x_s) \Rightarrow w^t \in \operatorname{span}(x_1, \ldots, x_s) (\forall t)$

Convergence
$\exists w, \|w\| = 1$, that $\gamma[w](S) = \gamma > 0 \Rightarrow w^t \cdot w \geq t\gamma$.
$R = \max_{x \in S} \|x\| \Rightarrow \|w^t\| \leq R\sqrt{t}$
$\cos \angle(u, w^t) = \frac{u \cdot w^t}{\|w^t\|} \geq \frac{t\gamma}{\sqrt{t}R} = \frac{\sqrt{t}\gamma}{R} \leq 1 \Rightarrow t \leq \frac{R^2}{\gamma^2}$

Covers Theorem
$C(s+1, n) = 2\sum_{i=0}^{n-1} \binom{s}{i}$
$C(S, n)$: Number of ways to separate S with n dimensions. $C(s, n) = 2s$ for $s \leq n$
Phase transition at $s = 2n$. For $s > 2n$ empty version space is the exception.

1.2 Hopfield Networks

Hopfield Model
$E(X) = -\frac{1}{2}\sum_{i \neq j} w_{ij} X_i X_j + \sum_i b_i X_i$ where $X_i \in \{-1, +1\}$
$w_{ij} = w_{ji}$ $(\forall i, j)$, $w_{ii} = 0$ $(\forall i)$: Interaction strengths

Hebbian Learning
$x^t \in \{\pm 1\}^n$ $(1 \leq t \leq s)$, $w_{ij} = \frac{1}{n}\sum_{t=1}^s x_i^t x_j^t$,
$W = \frac{1}{n}\sum_{t=1}^s x^t (x^t)^\top$

2. Feedforward Networks

2.1 Linear Models

Linear regression
$h[w](S) = \frac{1}{2s}\|Xw - y\|^2$, $\nabla h = 2X^\top X w - 2X^\top y$

Moore-Penrose inverse solution
$w^* = X^+ y \in \operatorname{argmin}_w h[w]$ where $X^+ := \lim_{\epsilon \to 0}(X^\top X + \epsilon I)^{-1} X^\top$

SGD update
$w_{t+1} := w_t + \eta \underbrace{(y_{i_t} - w_t^\top x_{i_t})}_{\text{residual}} x_{i_t}$
with $i_t \overset{\text{iid}}{\sim} \operatorname{Uniform}(1, \ldots, s)$

Gaussian noise model
$y_i = w^\top x_i + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. Least squares $\equiv$ neg log likelihood of noise model.

Ridge regression
$h_\lambda[w] := h[w] + \frac{\lambda}{2}\|w\|^2$, $w^* = (X^\top X + \lambda I)^{-1} X^\top y$

Logistic function
$\sigma(z) = \frac{1}{1+e^{-z}}$, $\sigma(z) + \sigma(-z) = 1$
$\sigma' = \sigma(1-\sigma)$, $\sigma'' = \sigma(1-\sigma)(1-2\sigma)$

Cross entropy loss
$\ell(y, z) = -y \log \sigma(z) - (1-y)\log(1 - \sigma(z))$
$= -\log \sigma((2y-1)z)$

Logistic regression gradient
$\nabla \ell_i = [\sigma(w^\top x_i) - y_i] x_i$

2.2 Feedforward Networks

Generic feedforward layer
$F: \underbrace{\mathbb{R}^{m(n+1)}}_{\text{parameters}} \times \underbrace{\mathbb{R}^n}_{\text{input}} \to \underbrace{\mathbb{R}^m}_{\text{output}}$,
$F[\theta](x) := \phi(Wx + b)$, $\theta := \operatorname{vec}(W, b)$

Composition of layers
$G = F^L[\theta^L] \circ \cdots \circ F^1[\theta^1]$
where $F^l[W^l, b^l](x) := \phi^l(W^l x + b^l)$

Layer activations
$x^l := (F^l \circ \cdots \circ F^1)(x) = F^l(x^{l-1})$, $x^0 = x$, $x^L = F(x)$

Softmax function
$\operatorname{softmax}(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$, $\operatorname{softmax}(A)_{ij} = \frac{e^{A_{ij}}}{\sum_k e^{A_{ik}}}$
$\ell(y; z) = \left[-zy + \log \sum_j e^{z_j}\right] \frac{1}{\ln 2}$

Residual layer
$F[W, b](x) = x + [\phi(Wx + b) - \phi(0)]$, therefore $F[0, 0] = \mathrm{id}$

Skip connection
Concatenate previous layer back in

2.3 Sigmoid Networks

Sigmoid/Tanh activations
$\sigma(z) = \frac{1}{1+e^{-z}}$, $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = 2\sigma(2z) - 1$,
$\tanh'(z) = 1 - \tanh^2(z)$

Barron's Theorem
For f with finite $C_f := \int \|\omega\| |\hat{f}(\omega)| d\omega$, $\exists$ MLP g with one hidden layer of width m: $\int_B (f - g_m)^2 \mu(dx) \leq \mathcal{O}(1/m)$

2.4 ReLU Networks

ReLU activation
$\phi(z) := (z)_+ := \max\{0, z\}$. ReLU networks are universal function approximators.

Zaslavsky: Connected regions
$R(\mathcal{H}) \leq \sum_{i=0}^{\min\{n,m\}} \binom{m}{i} := R(m)$

Montufar: Connected regions
$R(m, L) \geq R(m) \left\lfloor \frac{m}{n} \right\rfloor^{n(L-1)}$ (L: layers, m: width)

3. Gradient-Based Learning

3.1 Backpropagation

Parameter derivatives
$\frac{\partial x_i^l}{\partial w_{ij}^l} = \dot{\phi}_i^l x_j^{l-1}$, $\frac{\partial x_i^l}{\partial b_i^l} = \dot{\phi}_i^l$ where $\dot{\phi}_i^l := \phi'^l((w_i^l)^\top x^{l-1} + b_i^l)$

Loss derivatives
$\frac{\partial h}{\partial w_{ij}^l} = \delta_i^l \dot{\phi}_i^l x_j^{l-1}$, $\frac{\partial h}{\partial b_i^l} = \delta_i^l \dot{\phi}_i^l$ with $\delta_i^l = \frac{\partial h}{\partial x_i^l} \dot{\phi}_i^l$

3.2 Gradient Descent

GD update & flow
$\theta_{t+1} = \theta_t - \eta \nabla h(\theta_t)$, ODE: $\frac{d\theta}{dt} = -\nabla h(\theta)$

L-smoothness
$\|\nabla h(\theta_1) - \nabla h(\theta_2)\| \leq L\|\theta_1 - \theta_2\|$ $(\forall \theta_1, \theta_2)$
$\lambda_{\max}(\nabla^2 h) \leq L$, $\ell''(x) \leq L$
$\ell(w) - \ell(w') \leq \nabla \ell(w')^\top (w - w') + \frac{L}{2}\|w - w'\|_2^2$

Polyak-Łojasiewicz
$\frac{1}{2}\|\nabla h(\theta)\|^2 \geq \mu(h(\theta) - \min h)$ $(\forall \theta)$

Convergence rate
$\eta = 1/L \Rightarrow t = \frac{2L}{\epsilon^2}(h(\theta^0) - \min h)$ for ϵ-critical. With PL: $h(\theta^t) - \min h \leq (1 - \frac{\mu}{L})^t (h(\theta^0) - \min h)$

3.3 Acceleration and Adaptivity

Heavy ball momentum
$\theta_{t+1} = \theta_t - \eta \nabla h(\theta_t) + \beta(\theta_t - \theta_{t-1})$

Nesterov acceleration
$\tilde{\theta}_{t+1} = \theta_t + \beta(\theta_t - \theta_{t-1})$, $\theta_{t+1} = \tilde{\theta}_{t+1} - \eta \nabla h(\tilde{\theta}_{t+1})$

AdaGrad
$\theta_{t+1}^i = \theta_t^i - \eta_t^i \partial_i h(\theta^t)$, $\nu_t^i = \nu_{t-1}^i + [\partial_i h]^2$, $\eta_t^i = \frac{\eta}{\sqrt{\nu_t^i + \epsilon}}$

Adam
$g_t^i = \beta g_{t-1}^i + (1-\beta)\partial_i h$, $\nu_t^i = \alpha \nu_{t-1}^i + (1-\alpha)[\partial_i h]^2$,
$\theta_{t+1}^i = \theta_t^i - \frac{\eta}{\sqrt{\nu_t^i + \epsilon}} g_t^i$

RMSprop
Adam without momentum term (set $\beta = 0$)

3.4 SGD

SGD update & variance
$\theta_{t+1} = \theta_t - \eta \nabla h(\theta^t)(x_{i_t}, y_{i_t})$, $V[\theta] = \frac{1}{s}\sum_{i=1}^s \|\nabla h(S) - \nabla h(x_i, y_i)\|^2$

SGD convergence
General: $O(1/\sqrt{t})$, Strongly convex: $O(\log t / t)$, Additionally smooth: $O(1/t)$

3.5 Function properties

Convexity
$\ell(\lambda w + (1-\lambda)w') \leq \lambda \ell(w) + (1-\lambda)\ell(w')$, $\ell''(x) \geq 0$ $\forall x$
$\ell(w) \geq \ell(w') + \nabla \ell(w')^\top (w - w')$ (differentiable case)

Strong convexity
$\ell(w) \geq \ell(w') + \nabla \ell(w')^\top (w - w') + \frac{\mu}{2}\|w - w'\|_2^2$, $\ell''(x) \geq \mu$

4. Convolutional Networks

4.1 Convolutions

Convolution definition
$(f * g)(u) := \int_{-\infty}^{\infty} g(u - t) f(t) dt = \int_{-\infty}^{\infty} f(u - t) g(t) dt$

Fourier property
$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g)$

Discrete convolution
$(f * g)[u] := \sum_{t=-\infty}^{\infty} f[t] g[u - t]$

Cross-correlation
$(g \star f)[u] := \sum_{t=-\infty}^{\infty} g[t] f[u + t]$

Toeplitz matrices
$(f * g) = \text{Toeplitz-Matrix}(g) f$

4.2 Convolutional Networks

Conventions
Padding: Add zeros around input. Stride: Step size of convolution.

Max-Pooling
Take maximum value in windows (size r)

ConvNets for Images
$y[r][s, t] = \sum_u \sum_{s', t'} w[r, u][s', t'] \cdot x[u][s + s', t + t']$
(r: output channel, u: input channel)

Parameters count
$D = \#r \cdot \#u \cdot \#s' \cdot \#t'$ (channels $\times$ window size)

4.3 NLP with ConvNets

Word embedding
$\omega \mapsto x_\omega \in \mathbb{R}^n$

Conditional log-bilinear model
$P(\nu | \omega) = \frac{\exp[x_\omega^\top y_\nu]}{\sum_\mu \exp[x_\omega^\top y_\mu]}$
$h(\{x_\omega\}, \{y_\nu\}) = \sum_{(\omega,\nu)} \ell_{\omega,\nu}$, $\ell_{\omega,\nu} = -x_\omega^\top y_\nu + \ln \sum_\mu \exp[x_\omega^\top y_\mu]$

Negative sampling
$\tilde{\ell}_{\omega,\nu} = -\ln \sigma(x_\omega^\top y_\nu) - \beta \mathbb{E}_{\mu \sim D} \ln(1 - \sigma(x_\omega^\top y_\mu))$

5. Recurrent Networks

5.1 Simple RNNs

Time evolution
$z_t := F[\theta](z_{t-1}, x_t)$, $z_0 := 0$ $(\forall t)$

Output map
$\hat{y}_t := G[\psi](z_t)$

RNN parameterization
$F[U, V](z, x) := \varphi(Uz + Vx)$, $G[W](z) := \Phi(Wz)$, $W \in \mathbb{R}^{q \times m}$

BPTT
$\frac{\partial h}{\partial z_i^t} = \sum_{s=t}^T \sum_{k=1}^m \sum_j \frac{\partial \bar{y}_k^s}{\partial z_j^s} \frac{\partial z_j^s}{\partial z_i^t}$, $\frac{\partial \bar{y}_k^s}{\partial z_j^s} = \dot{\Phi}_k^s w_{kj}$
$\frac{\partial h}{\partial v_{ij}} = \sum_{t=1}^T \frac{\partial h}{\partial z_i^t} \dot{\varphi}_i^t x_j^t$, $\frac{\partial h}{\partial u_{ij}} = \sum_{t=1}^T \frac{\partial h}{\partial z_i^t} \dot{\varphi}_i^t z_j^{t-1}$

Spectral norm
$\|A\|_2 = \max_{x:\|x\|=1} \|Ax\|_2 = \sigma_1(A)$

Gradient norms
$\frac{\partial z_T}{\partial z_0} = \dot{\Phi}^T U \cdots \dot{\Phi}^1 U$. Vanishes if $\sigma_1(U) < 1/\kappa$, explodes if $\sigma_1(U)$ too large.

Bidirectional RNNs
$\hat{y}_t = \Phi(W z_t + \tilde{W} \tilde{z}_t)$

5.2 Gated Memory

LSTM
$z_t := \sigma(F\tilde{x}_t) \odot z_{t-1} + \sigma(G\tilde{x}_t) \odot \tanh(V\tilde{x}_t)$, $\tilde{x}_t := [x_t, \ell_t]$,
$\ell_{t+1} = \sigma(H\tilde{x}_t) \odot \tanh(Uz_t)$

GRU
$z_t = (1 - \sigma) \odot z_{t-1} + \sigma \odot \tilde{z}_t$, $\sigma := \sigma(G[x_t, z_{t-1}])$
$\tilde{z}_t := \tanh(V[\ell_t \odot z_{t-1}, x_t])$, $\ell_t := \sigma(H[z_{t-1}, x_t])$

5.3 Linear Recurrent Models

Linear state evolution
$z_{t+1} = Az_t + Bx_t$

Diagonal form
$A = P\Lambda P^{-1}$, $\Lambda := \operatorname{diag}(\lambda_1, \ldots, \lambda_m)$, $\lambda_i \in \mathbb{C}$

Stability

$\max_j |\lambda_j| \le 1$

Initialization

$\lambda_i = \exp(-\exp(\nu_i) + i\theta_i)$, $e^{\nu_i} = -\ln r_i$
$\theta_i \sim \mathrm{Uni}[0;2\pi]$, $r_i \sim \mathrm{Uni}[I]$, $I \subseteq [0;1]$

Advantages

(i) clear long/short range dependencies (ii) no channel mixing required (iii) parallelizable training

6. Attention and Transformers

6.1 Attention

Attention mixing

$\xi_s := \sum_t a_{st} W x_t$, $a_{st} \ge 0$, $\sum_t a_{st} = 1$
$A = (a_{st}) \in \mathbb{R}^{T\times T}$, $\Xi = WXA^\top$

Query-key matching

$Q = U_Q X$, $K = U_K X$ ($U_Q, U_K \in \mathbb{R}^{q\times n}$)
$Q^\top K = X^\top U_Q^\top U_K X$ ($Q^\top K \in \mathbb{R}^{T\times T}$, rank $\le q$)

Softmax attention

$A = \mathrm{softmax}(\beta Q^\top K)$, $a_{st} = \frac{e^{\beta[Q^\top K]_{st}}}{\sum_r e^{\beta[Q^\top K]_{sr}}}$, usually $\beta = 1/\sqrt{q}$

Feature transformation

$X \mapsto Y \mapsto F(Y)$, $F(\theta)(Y) = (F(y_1), \dots, F(y_T))$

Positional encoding

$p_{tk} = \begin{cases} \sin(t\omega_k) & k \text{ even} \\ \cos(t\omega_k) & k \text{ odd} \end{cases}$, $\omega_k = C^{k/K}$

Transformer architecture

Self-attention: attend to its own values in the past.
Cross-attention: decoder attends to encoder output.

Vision transformer patch embedding

$\mathbb{R}^{p\times p\times c} \ni \mathrm{patch}_t \mapsto x_t := V\vec{(\mathrm{patch}_t)} \in \mathbb{R}^n$
$V \in \mathbb{R}^{n\times(cp^2)}$

GELU activation

$\varphi(z) = z\Pr(z \le Z)$, $Z \sim \mathcal{N}(0,1)$

7. Geometric Deep Learning

7.1 Sets and Points

Order-invariance

$f(x_1, \dots, x_M) = f(x_{\pi_1}, \dots, x_{\pi_M})\ \forall \pi \in S_M$
Permutation invariant sum: $\sum_{m=1}^M x_m = \sum_{m=1}^M x_{\pi_m}$, $\forall M$, $\forall \pi \in S_M$

Equivariance

$f(x_{\pi_1}, \dots, x_{\pi_M}) = (y_{\pi_1}, \dots, y_{\pi_M})$

Deep Sets model

$f(x_1, \dots, x_M) = \rho\left(\sum_{m=1}^M \varphi(x_m)\right)$

Max pooling variant

$f(x_1, \dots, x_M) = \rho\left(\max_{m=1}^M \varphi(x_m)\right)$

Equivariant map

$\rho : \mathbb{R} \times \mathbb{R}^N \to Y$, $(x_m, \sum_{k=1}^M \varphi(x_k)) \mapsto y_m$

7.2 Graph Conv Networks

Feature & adjacency

$X = [x_1^\top; \dots; x_M^\top]$, $A = (a_{nm})$ with $a_{nm} = 1$ if $\{v_n, v_m\} \in E$

Graph invariance

$f(X, A) = f(PX, PAP^\top)\ \forall P$

Graph equivariance

$f(X, A) = Pf(PX, PAP^\top)\ \forall P$

Message passing

$\varphi(x_m, X_m) = \varphi(x_m, \bigoplus_{X_m} \Phi(x))$, $\bigoplus$ permutation-invariant

Normalized adjacency

$\bar{A} = D^{-1/2}(A + I)D^{-1/2}$, $D = \mathrm{diag}(d_m)$, $d_m = 1 + \sum_n a_{nm}$

GCN layer

$X^+ = \sigma(\bar{A}XW)$, $W \in \mathbb{R}^{M\times N}$
Two-layer GCN: $Y = \mathrm{softmax}\left(\bar{A}\sigma\left(\bar{A}XW^{(0)}\right)W^{(1)}\right)$

7.3 Spectral Graph Theory

Laplacian operator

$\Delta f := \sum_{n=1}^N \frac{\partial^2 f}{\partial x_n^2}$, $f : \mathbb{R}^N \to \mathbb{R}$

Graph Laplacian

$L = D - A$, $(Lx)_n = \sum_m a_{nm}(x_n - x_m)$

Normalized Laplacian

$\tilde{L} = I - D^{-1/2}AD^{-1/2}$

Graph Fourier transform

$L = U\Lambda U^\top$, $\Lambda := \mathrm{diag}(\lambda_1, \dots, \lambda_M)$, $\lambda_i \ge \lambda_{i+1}$
Convolution: $x * y = U((U^\top x) \odot (U^\top y))$
Filtering: $G_\theta(L)x = UG_\theta(\Lambda)U^\top x$

Polynomial kernels

$U\left(\sum_{k=0}^K \alpha_k \Lambda^k\right)U^\top = \sum_{k=0}^K \alpha_k L^k$
Polynomial kernel network layer: $x_i^{l+1} = \sum_j p_{ij}(L)x_j^l + b_i$, $p_{ij}(L) = \sum_{k=0}^K \alpha_{ijk} L^k$

7.4 Attention GNNs

Attention coupling

$q_{ij} = \mathrm{softmax}(f^l(u^\top(Vx_i; Vx_j; x_{ij})))$ s.t. $\sum_j A_{ij} q_{ij} = 1$

Attention propagation

$X^+ = \sigma(QXW)$

Weisfeiler-Lehman test

Graph isomorphism test: iteratively aggregate node labels from neighborhoods.

8. Tricks of the Trade

8.1 Initialization

Random initialization

$\theta_i^0 \sim \mathcal{N}(0, \sigma_i^2)$ or $\theta_i^0 \sim \mathrm{Uniform}[-\sqrt{3}\sigma_i; \sqrt{3}\sigma_i]$

LeCun initialization

$w_{ij} \sim \mathrm{Uniform}[-a; a]$, $a := 1/\sqrt{n}$, $b_i = 0$. Stabilizes variance.

Glorot initialization

$w_{ij} \sim \mathrm{Uniform}[-\sqrt{3\epsilon}; \sqrt{3\epsilon}]$, $\epsilon := 2/(n+m)$. Stabilizes gradient variance.

He initialization

$w_{ij} \sim \mathcal{N}(0, \epsilon)$ or $\mathrm{Uniform}[-\sqrt{3\epsilon}; \sqrt{3\epsilon}]$, $\epsilon := 2/n$. For ReLU (half units active).

Orthogonal initialization

$\frac{1}{\sqrt{m}}W \sim \mathrm{Uniform}(O(m))$ s.t. $W^\top W = WW^\top = mI$

8.2 Weight Decay

L_2 regularization

$\mu(\theta) = \frac{\mu}{2}\|\theta\|^2$

GD with weight decay

$\dot{\theta} = -\eta\nabla E(\theta) - \eta\mu\theta$

Local loss landscape

$\theta_\mu^* = (H + \mu I)^{-1}H\theta^*$. Minimum shrunk along small eigenvalue directions.

Optimal weight decay

$\mu = \sigma^2/u^2$. Inverse proportional to signal-to-noise ratio.

8.3 Dropout

Dropout as Ensembling

$p(y|x) = \sum_{b\in\{0,1\}^R} p(b)p(y|x;b)$, $p(b) = \prod_{i=1}^R \pi_i^{b_i}(1 - \pi_i)^{1-b_i}$

Weight scaling for inference

$\tilde{w}_{ij} \leftarrow \pi_j w_{ij}$

8.4 Normalization

Batch normalization

$\bar{f} = \frac{f - \mathbb{E}[f]}{\sqrt{\mathbb{V}[f]}}$, $\mathbb{E}[\bar{f}] = 0$, $\mathbb{V}[\bar{f}] = 1$
$\bar{f}[\gamma, \beta] = \gamma + \beta\bar{f}$ (learnable)

Weight normalization

$f(v, \epsilon)(x) = \varphi(w^\top x)$, $w := \frac{\epsilon}{\|v\|_2}v$
$\partial_\epsilon E = \nabla_w E \cdot \frac{v}{\|v\|_2}$, $\nabla_v E = \frac{\epsilon}{\|v\|_2}\left(I - \frac{ww^\top}{\|w\|_2^2}\right)\nabla_w E$

Layer normalization

$\bar{f}_i = \frac{f_i - \mathbb{E}[f]}{\sqrt{\mathbb{V}[f]}}$, $\mathbb{E}[f] = \frac{1}{m}\sum_i f_i$, $\mathbb{V}[f] = \frac{1}{m}\sum_i (f_i - \mathbb{E}[f])^2$

Reparameterization trick

$z = \mu + \Sigma^{1/2}\eta$, $\eta \sim \mathcal{N}(0, I)$
$\nabla_\mu \mathbb{E}[f(z)] = \mathbb{E}[\nabla_z f(z)]$, $\nabla_\Sigma \mathbb{E}[f(z)] = \frac{1}{2}\mathbb{E}[\nabla_z^2 f(z)]$

8.5 Model Distillation

Tempered cross entropy

$\ell(x) = \sum_y \frac{q\exp[F_y/T]}{\sum_\nu \exp[F_\nu/T]}[\frac{1}{T}G_y - \ln\sum_\nu \exp[G_\nu/T]]$

Distillation gradient

$\frac{\partial \ell}{\partial G_y} = \frac{1}{T}[\frac{e^{F_y/T}}{\sum_\nu e^{F_\nu/T}} - \frac{qe^{G_y/T}}{\sum_\nu e^{G_\nu/T}}]$

9. Theory

9.1 Infinite Width (NTK)

Neural tangent kernel

$k(x, \xi) = \nabla f(x) \cdot \nabla f(\xi)$, $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$
Linearized: $h(\beta)(x) = f(x) + \beta \cdot \nabla f(x)$ with $\beta \approx \theta - \theta_0$

Gradient flow

ODE: $\dot{\theta} = \sum_{i=1}^s (y_i - f_i(\theta))\nabla f_i(\theta)$
Functional: $\dot{f}_j = \sum_{i=1}^s (y_i - f_i)k^{(\theta)}(x_i, x_j)$, $\dot{f} = K^{(\theta)}(y - f)$

Dual representation

$h(\alpha)(x) = f(x) + \sum_{i=1}^s \alpha_i \nabla f(x_i) \cdot \nabla f(x)$
Optimal: $\alpha^* = K^+(y - f)$, $h^*(x) = k(x)K^+(y - f)$

Infinite width limit

$w_{ij}^l = \frac{\sigma_w}{\sqrt{m}}\omega_{ij}^l$, $b_i^l = \frac{\sigma_b}{\sqrt{m}}\beta_i^l$, $\omega^l, \beta^l \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$
$k^{(\theta)} \to k^\infty$ for $m_l \to \infty$

NTK constancy

$\frac{dk^{(\theta(t))}}{dt} = 0$, $f^\infty(x) = k(x)K^+(y - f)$
Near-constancy: $\|k(\theta_0) - k(\theta_t)\|_F^2 \in O(1/m)$

Kernel regression solution

$\bar{F}_\infty = K_0(K_0 + \lambda I)^{-1}Y$

Function space

$\bar{F} \in \mathcal{H}_K$ (RKHS), $\|\bar{F}\|_{\mathcal{H}_K}^2 = \theta^\top\theta$

9.2 Bayesian DNNs

Parameter prior

$p(\theta) = \prod_{i=1}^d p(\theta_i)$, $\theta_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$
$-\log p(\theta) = \frac{1}{2\sigma^2}\|\theta\|^2 + \text{const}$ (weight decay)

Likelihood

$y_i = f^*(x_i) + \eta_i$, $\eta_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \delta^2)$
$-\log p(S|\theta) = \frac{1}{2\delta^2}\|y - f(\theta)\|^2 + \text{const}$

Posterior

$p(\theta|S) = \frac{p(\theta)p(S|\theta)}{p(S)}$, $-\log p(\theta|S) = E(\theta) + \text{const}$
$E(\theta) = \frac{1}{2\delta^2}\|y - f\|^2 + \frac{1}{2\sigma^2}\|\theta\|^2$

Predictive distribution

$\bar{f}(x) = \int f(\theta)(x)p(\theta|S)d\theta$
Bayesian ensembling: $\bar{f}^{(n)}(x) = \frac{\sum_{i=1}^n \exp[-E(\theta_i)]f(\theta_i)(x)}{\sum_{j=1}^n \exp[-E(\theta_j)]}$

9.3 GPs & Infinite Width

Gaussian processes

$(f(x_1), \dots, f(x_s)) \sim \mathcal{N}$, $\sum_{i=1}^s \alpha_i f(x_i) \sim \mathcal{N}\ \forall\alpha \in \mathbb{R}^s$
Mean $\mu(x) := \mathbb{E}_x[f(x)]$, covariance $k(x, \xi) := \mathbb{E}_{x,\xi}[f(x)f(\xi)] - \mu(x)\mu(\xi)$

GPs in DNNs

Linear layer: $w \sim \mathcal{N}(0, \frac{\sigma^2}{n}I)$, $\mathbb{E}[y_i y_j] = \sigma^2 x_i^\top x_j$
Deep layers: near-normal for high-dim inputs. $f \sim \mathcal{GP}(0, K^{l-1})$

Kernel recursion

$K_{\mu\nu}^l = \mathbb{E}[\varphi(x_{i\mu}^{l-1})\varphi(x_{i\nu}^{l-1})] = \sigma^2\mathbb{E}[\varphi(f_\mu)\varphi(f_\nu)]$
Example kernels: $k(x, \xi) = x^\top\xi$, $k(x, \xi) = e^{-\gamma\|x - \xi\|^2}$

Kernel regression

$f^*(x) = k(x)^\top K^+ y$ (mean of Bayesian predictive)
$\mathbb{E}[(f(x) - f^*(x))^2] = K(x, x) - k(x)^\top K^+ k(x)$

9.4 Statistical Learning Theory

Generalization gap

$\mathcal{R}(f) - \hat{\mathcal{R}}(f) = \mathbb{E}_{x,y}[\ell(f(x), y)] - \frac{1}{n}\sum_i \ell(f(x_i), y_i)$

PAC bound

$\mathbb{P}[\mathcal{R}(f) - \hat{\mathcal{R}}(f) \le \epsilon] \ge 1 - \delta$

VC dimension

VC-dim$(F) := \max_s \sup_{|S|=s} \mathbf{1}[|F(S)| = 2^s]$
VC inequality: $\mathbb{P}[\sup_F |\hat{E}(f) - E(f)| > \epsilon] \le 8|F(s)|e^{-s\epsilon^2/32}$

Rademacher complexity

$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_{\sigma,S}[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_i \sigma_i f(x_i)]$

Generalization bound

$\mathcal{R}(f) \le \hat{\mathcal{R}}(f) + 2\mathcal{R}_n(\mathcal{F}) + \sqrt{\frac{\ln(1/\delta)}{2n}}$

Bias-variance tradeoff

$\mathbb{E}[(f(x)-y)^2] = \text{Bias}^2 + \text{Var} + \sigma^2$

Generalization gap

$\Delta := \max(0, E - \hat{E})$, E: expected population error, $\hat{E}$: empirical

Double descent

Beyond interpolation point, models may level out at lower generalization error.

KL divergence

$D_{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx = \mathbb{E}_{x\sim p}[\ln \frac{p(x)}{q(x)}]$

9.5 Loss Landscape

Critical points

$\nabla\mathcal{L}(\theta^*) = 0$. Local min: $H \succeq 0$. Saddle: H indefinite.

Sharpness

$\lambda_{\max}(H)$. Flat minima $\to$ better generalization.

Mode connectivity

Local minima connected by paths of low loss.

Lottery ticket hypothesis

Sparse subnetworks can match full network performance if initialized correctly.

10. Generative Models

10.1 Variational Auto Encoders

Linear autoencoder

$x \mapsto z = Cx$, $z \mapsto \hat{x} = Dz$, $E(C,D)(x) = \frac{1}{2}\|x - DCx\|^2$

$DCX = \hat{X} = U\Sigma_m V^\top$, for centered data $\equiv$ PCA

Linear factor analysis

$x = \mu + Wz + \eta$, $\eta \sim \mathcal{N}(0,\Psi)$, $x \sim \mathcal{N}(\mu, WW^\top + \Psi)$ for $z \sim \mathcal{N}(0, I)$

$\mu_{z|x} = W^\top(WW^\top + \Psi)^{-1}(x-\mu)$

Generative model

$p_\theta(x,z) = p_\theta(x|z)p(z)$, $p(z) = \mathcal{N}(0,I)$

ELBO

$\log p(\theta)(x) = \log \int q(z) p(\theta)(x|z) \frac{p(z)}{q(z)} dz$

$\ge \int q(z) \log p(\theta)(x|z) dz - D_{KL}(q\|p) =: L(\theta, q)(x)$

Inference network

$z \sim \mathcal{N}(\mu(x), \Sigma(x))$

Encoder

$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma^2_\phi(x))$

Decoder

$p_\theta(x|z) = \mathcal{N}(\mu_\theta(z), \sigma^2 I)$ or Bernoulli

Reparameterization trick

$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$, $\epsilon \sim \mathcal{N}(0,I)$

KL divergence (Gaussian)

$D_{KL} = \frac{1}{2}\sum_j(\mu_j^2 + \sigma_j^2 - \ln\sigma_j^2 - 1)$

β-VAE

$\mathcal{L} = \mathbb{E}_q[\ln p(x|z)] - \beta D_{KL}(q\|p)$. $\beta > 1$ for disentanglement.

10.2 Generative Adversarial Networks

Generator

$G : z \mapsto x$, $z \sim p(z)$

Discriminator

$D : x \mapsto [0,1]$, probability that x is real

GAN objective

$V(G,D) = \mathbb{E}_{x_r\sim p_{\text{data}}}[D(x_r)] + \mathbb{E}_{z\sim p_z}[1 - D(G(z))]$

Bayes-optimal classifier

$q_\theta(x) := P\{y=1|x\} = \frac{p(x)}{p(x)+p_\theta(x)}$

Jensen-Shannon objective

$\ell^* = \text{JS}(p, p_\theta) - \ln 2$

$\ell^*(\theta) \ge \sup_\phi \ell(\theta,\phi)$ where ϕ: discriminator, θ: generator

Alternating gradient descent

$\theta_{t+1} = \theta_t - \eta\nabla_\theta \ell(\theta_t, \phi_t)$

$\phi_{t+1} = \phi_t + \eta\nabla_\phi \ell(\theta^{t+1}, \phi_t)$

Wasserstein GAN

$\min_G \max_{D\in 1\text{-Lip}} \mathbb{E}_x[D(x)] - \mathbb{E}_z[D(G(z))]$

Gradient penalty (WGAN-GP)

$\lambda\mathbb{E}_{\hat{x}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]$, $\hat{x} = \alpha x + (1-\alpha)G(z)$

10.3 Denoising Diffusion

Forward process

$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$

Marginal

$q(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, \bar{\beta}_t I)$, $\bar{\alpha}_t = \prod_{\tau=1}^t(1-\beta_\tau)$, $\bar{\beta}_t = 1 - \bar{\alpha}_t$

$\nu_t \approx \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, \bar{\beta}_t I) \xrightarrow{t\to\infty} \mathcal{N}(0,I)$

Reverse process

$p_\theta(x_{t-1}|x_t) = \mathcal{N}(m_\theta(x_t,t), \Sigma(x_t,t))$

Forward: $\pi^* \to \nu_T = \pi$. Backward: $\pi \to \mu_0^\theta \approx \pi^*$

Training objective

$L_t = \frac{\|m(x_t,x_0,t) - m_\theta(x_t,t)\|^2}{2\sigma_t^2} + \text{const}$

Reparameterization: $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$

Simplified criterion

$h(\theta)(x) = \frac{1}{T}\sum_{t=1}^T \mathbb{E}[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2]$

Forward trajectory target

$x_{t-1}|x_t, x_0 = \mathcal{N}(m(x_t,x_0,t), \tilde{\beta}_t I)$

$m(x_t,x_0,t) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{(1-\bar{\alpha}_{t-1})\sqrt{1-\beta_t}}{1-\bar{\alpha}_t}x_t$

Sampling

$m(x_t,x_0,t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$

Score function

$\nabla_x \ln p(x) \approx -\frac{\epsilon_\theta(x,t)}{\sqrt{1-\bar{\alpha}_t}}$

Classifier-free guidance

$\tilde{\epsilon} = (1+w)\epsilon_\theta(x_t,t,c) - w\epsilon_\theta(x_t,t)$

DDIM (deterministic)

$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta}{\sqrt{\bar{\alpha}_t}}) + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_\theta$

11. Ethics

11.1 Adversarial Examples

Adversarial perturbation

$x' = x + \delta$, $\|\delta\|_p \le \epsilon$, $F(x') \ne F(x)$

$\|x\|_p = (\sum_i |x_i|^p)^{1/p}$, $\|x\|_\infty = \max_i |x_i|$, $\|x\|_0 = |\{i : x_i \ne 0\}|$

FGSM (Fast Gradient Sign Method)

$\delta = \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f(x), y))$

PGD (Projected Gradient Descent)

$\eta^{t+1} = \Pi_\epsilon\left[\eta^t + \alpha\nabla_x\mathcal{L}(f(x+\eta^t), y)\right]$

$\Pi_\epsilon[z] := z/\|z\|_2$ for $p=2$, $\Pi_\epsilon[z] := z/\|z\|_\infty$ for $p=\infty$

Adversarial training

$\min_\theta \mathbb{E}_{x,y}[\max_{\|\delta\|\le\epsilon} \mathcal{L}(f(x+\delta), y; \theta)]$

DeepFool (binary)

Optimal perturbation: $\eta \propto \text{sign}(f_1(x) - f_2(x))(w_2 - w_1)$ for $f_i = w_i^\top x + b_i$. Iterate: $\arg\min_{\|\eta\|_2}$ s.t. $(\nabla f_1 - \nabla f_2)^\top \eta < f_1(x) - f_2(x)$

Certified robustness

Provable guarantees via randomized smoothing, interval bound propagation.

Transferability

Adversarial examples often transfer between models.