# Lecture 14: VC Dimension, Validation, Neural Networks, Kernels (workshop)

Nisha Chandramoorthy

October 5, 2023

# VC dimension

- A hypothesis class restricted to some $C$ *shatters* $C$ if any binary function on $C$ is in the class

# VC dimension

- A hypothesis class restricted to some $C$ *shatters* $C$ if any binary function on $C$ is in the class
- VC dimension, $\mathrm{VCdim}(\mathcal{H})$: maximal size of $C \subseteq \mathcal{X}$ that is shattered by $\mathcal{H}$.

# VC dimension

- A hypothesis class restricted to some $C$ *shatters* $C$ if any binary function on $C$ is in the class
- VC dimension, $\mathrm{VCdim}(\mathcal{H})$: maximal size of $C \subseteq \mathcal{X}$ that is shattered by $\mathcal{H}$.
- Eg 1. VCdim of threshold functions on $\mathbb{R}$ is 1.

# VC dimension

- A hypothesis class restricted to some $C$ *shatters* $C$ if any binary function on $C$ is in the class
- VC dimension, $\mathrm{VCdim}(\mathcal{H})$: maximal size of $C \subseteq \mathcal{X}$ that is shattered by $\mathcal{H}$.
- Eg 1. VCdim of threshold functions on $\mathbb{R}$ is 1.
- Eg 2: VCdim of indicator functions on intervals of $\mathbb{R}$ is 2.

# VC dimension

- A hypothesis class restricted to some $C$ *shatters* $C$ if any binary function on $C$ is in the class
- VC dimension, $\mathrm{VCdim}(\mathcal{H})$: maximal size of $C \subseteq \mathcal{X}$ that is shattered by $\mathcal{H}$.
- Eg 1. VCdim of threshold functions on $\mathbb{R}$ is 1.
- Eg 2: VCdim of indicator functions on intervals of $\mathbb{R}$ is 2.
- Eg 3: VCdim of a finite class $\mathcal{H} \leqslant \log_2 |\mathcal{H}|$

# Generalization bounds based on VC dimension

- $\mathcal{H} = \{h_\theta(x) = \sin(\theta x) : \theta \in \mathbb{R}\}$.

# Generalization bounds based on VC dimension

- $\mathcal{H} = \{h_\theta(x) = \sin(\theta x) : \theta \in \mathbb{R}\}$.
- VCdim is $\infty$

# Generalization bounds based on VC dimension

- $\mathcal{H} = \{h_\theta(x) = \sin(\theta x) : \theta \in \mathbb{R}\}$.
- VCdim is $\infty$
- Binary classification generalization for 0-1 loss over class $\mathcal{H}$ with VCdim = $d$: there exist constants $C_1, C_2 > 0$ such that

$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leqslant m_{\mathcal{H}}(\epsilon, \delta) \leqslant C_2 \frac{d + \log(1/\delta)}{\epsilon}$$

- Consistent with bound from lecture 2.

- Consistent with bound from lecture 2.
- $m_{\mathcal{H}} = \dfrac{\log(|\mathcal{H}|/\delta)}{\epsilon}$
- $\mathcal{D}^m(S : R(h_S) \geqslant \epsilon) \leqslant |\mathcal{H}_b| e^{-\epsilon m} \leqslant |\mathcal{H}| e^{-\epsilon m}$.

- Consistent with bound from lecture 2.
- $m_{\mathcal{H}} = \dfrac{\log(|\mathcal{H}|/\delta)}{\epsilon}$
- $\mathcal{D}^m(S : R(h_S) \geqslant \epsilon) \leqslant |\mathcal{H}_b| e^{-\epsilon m} \leqslant |\mathcal{H}| e^{-\epsilon m}$.

# VC dimension contd

# Cross Validation

- Recall Hoeffding's inequality: $X_1, \cdots, X_m$ iid sequence with $P(a \leqslant X \leqslant b) = 1$. Then, with probability $\geqslant 1 - \delta$,

$$|\frac{1}{m}\sum_{i=1}^{m} X_i - EX| < (b-a)\sqrt{\frac{\log(2/\delta)}{2m}}$$

# Cross Validation

- Recall Hoeffding's inequality: $X_1, \cdots, X_m$ iid sequence with $P(a \leqslant X \leqslant b) = 1$. Then, with probability $\geqslant 1 - \delta$,

$$|\frac{1}{m}\sum_{i=1}^{m} X_i - EX| < (b-a)\sqrt{\frac{\log(2/\delta)}{2m}}$$

- For any set $V$ of size $m$, when loss $\in (0,1)$, by Hoeffding's inequality,

$$|R_V(h) - R(h)| \leqslant \sqrt{\frac{\log(2/\delta)}{2m}}$$

- $k$-fold cross validation to choose models (e.g., regularization parameters):
  - Divide given set $S$ into $k$ subsets (folds)
  - For each parameter, each fold: run learning algorithm on union of all folds except one; calculate test/validation loss on fold
  - Run algorithm on $S$ using parameter with minimum total test/validation loss.

# Models/representations, algorithms, statistical principles

- how to make and test conjectures about how large language models (LLMs) learn?

# Models/representations, algorithms, statistical principles

- how to make and test conjectures about how large language models (LLMs) learn?
- "how to train them better (more efficiently)" – number of practical questions perhaps benefit from theory

# Models/representations, algorithms, statistical principles

- how to make and test conjectures about how large language models (LLMs) learn?
- "how to train them better (more efficiently)" – number of practical questions perhaps benefit from theory
- AI safety, fair and ethical ethical use, combining with other domain knowledge (e.g., physics, chemistry etc).... and many more!

# Models/representations, algorithms, statistical principles

- how to make and test conjectures about how large language models (LLMs) learn?
- “how to train them better (more efficiently)” – number of practical questions perhaps benefit from theory
- AI safety, fair and ethical ethical use, combining with other domain knowledge (e.g., physics, chemistry etc).... and many more!
- Perhaps biggest contribution advance to LLMs: transformers and their training.

# (partial) History - trace back from transformers (source:Wikipedia)

- Transformer architecture: 2017, Google Brain [Vaswani et al]
- Deep learning, unsupervised learning 2010s (e.g., GANs 2014)...
- ImageNet: 2009, Fei Fei Li
- Long-short term memory (LSTM) architecture: 1997, [Hochreiter and Schmidhuber]
- Convolutional NNs: (inspired from) 1979 work by [Fukushima]; Recurrent neural networks: 1982 [Hopfield]
- ...
- Automatic Differentiation: 1970 [Linnainmaa]
- ...
- First neural networks: 1950s [Minsky and others]

# Fully connected Neural Networks

- Neuron: input $\sum_j w_j h_j$; output $\sigma(\sum_j w_j h_j)$
- Organized into layers of depth $l$ and width $n$
- Graph: $V$, $E$, $\sigma$, $w$; weight function.

# Fully connected Neural Networks

- Neuron: input $\sum_j w_j h_j$; output $\sigma(\sum_j w_j h_j)$
- Organized into layers of depth $l$ and width $n$
- Graph: $V$, $E$, $\sigma$, $w$; weight function.
- VC dim of $\mathcal{H}_{V,E,\text{sign}} \leqslant C|E|\log|E|$

# Fully connected Neural Networks

- Neuron: input $\sum_j w_j h_j$; output $\sigma(\sum_j w_j h_j)$
- Organized into layers of depth $l$ and width $n$
- Graph: $V$, $E$, $\sigma$, $w$; weight function.
- VC dim of $\mathcal{H}_{V,E,\mathrm{sign}} \leqslant C|E|\log|E|$
  - Proof: Growth function $\tau_{\mathcal{H}}(m) = \max_{C,|C|=m} |\mathcal{H}|_C|$

# Fully connected Neural Networks

- Neuron: input $\sum_j w_j h_j$; output $\sigma(\sum_j w_j h_j)$
- Organized into layers of depth $l$ and width $n$
- Graph: $V$, $E$, $\sigma$, $w$; weight function.
- VC dim of $\mathcal{H}_{V,E,\mathrm{sign}} \leqslant C|E| \log |E|$
  - Proof: Growth function $\tau_{\mathcal{H}}(m) = \max_{C,|C|=m} |\mathcal{H}|_C|$ Fact 1: Let $\mathcal{H} = \mathcal{H}_l \circ \cdots \circ \mathcal{H}_1$. Then, $\tau_{\mathcal{H}}(m) \leqslant \prod_{t=1}^{l} \tau_{\mathcal{H}_t}(m)$.

# Fully connected Neural Networks

- Neuron: input $\sum_j w_j h_j$; output $\sigma(\sum_j w_j h_j)$
- Organized into layers of depth $l$ and width $n$
- Graph: $V$, $E$, $\sigma$, $w$; weight function.
- VC dim of $\mathcal{H}_{V,E,\mathrm{sign}} \leqslant C|E|\log|E|$
  - Proof: Growth function $\tau_{\mathcal{H}}(m) = \max_{C,|C|=m} |\mathcal{H}|_C|$ Fact 1: Let $\mathcal{H} = \mathcal{H}_l \circ \cdots \circ \mathcal{H}_1$. Then, $\tau_{\mathcal{H}}(m) \leqslant \prod_{t=1}^{l} \tau_{\mathcal{H}_t}(m)$.
  - Fact 2: Let $\mathcal{H} = \mathcal{H}^{(1)} \cdots \circ \mathcal{H}^{(n)}$. Then, $\tau_{\mathcal{H}}(m) \leqslant \prod_{t=1}^{l} \tau_{\mathcal{H}^{(t)}}(m)$.

# Fully connected Neural Networks

- Neuron: input $\sum_j w_j h_j$; output $\sigma(\sum_j w_j h_j)$
- Organized into layers of depth $l$ and width $n$
- Graph: $V$, $E$, $\sigma$, $w$; weight function.
- VC dim of $\mathcal{H}_{V,E,\mathrm{sign}} \leqslant C|E| \log |E|$
  - Proof: Growth function $\tau_{\mathcal{H}}(m) = \max_{C,|C|=m} |\mathcal{H}|_C|$ Fact 1: Let $\mathcal{H} = \mathcal{H}_l \circ \cdots \circ \mathcal{H}_1$. Then, $\tau_{\mathcal{H}}(m) \leqslant \prod_{t=1}^{l} \tau_{\mathcal{H}_t}(m)$.
  - Fact 2: Let $\mathcal{H} = \mathcal{H}^{(1)} \cdots \circ \mathcal{H}^{(n)}$. Then, $\tau_{\mathcal{H}}(m) \leqslant \prod_{t=1}^{l} \tau_{\mathcal{H}^{(t)}}(m)$.
  - Fact 3: Sauer's Lemma: $\tau_{\mathcal{H}}(m) = (em/d)^d$, where $d \geqslant$ VCdim $(\mathcal{H})$