Follow the Leader with Dropout Purturbations

Manfred K. Warmuth
UCSC

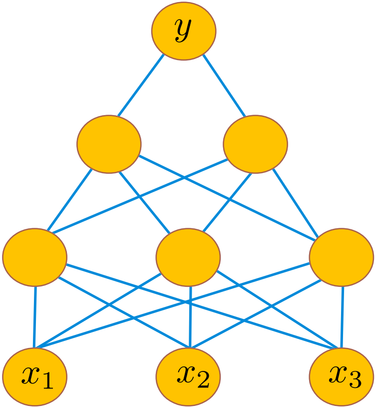
December 12, NIPS 2014 workshop on perturbations

Joint work with Tim Van Erven and Wojciech Kotłowski
Universiteit Leiden

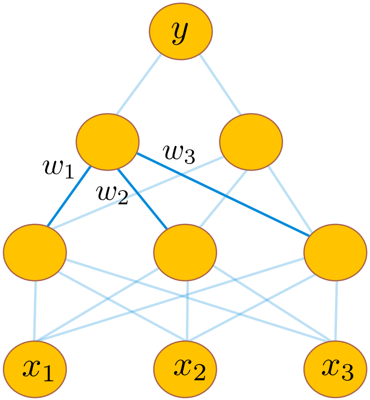
Major insights from [Devroye, Lugosi, Neu 2013]

Outline

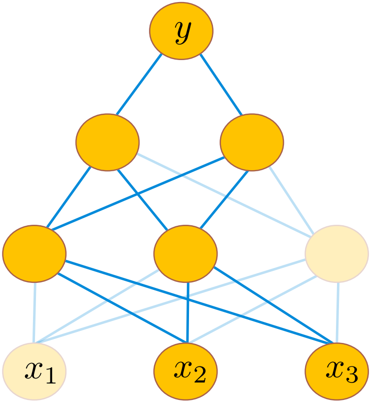
Feed forward neural net



Weights parameters - sigmoids at internal nodes



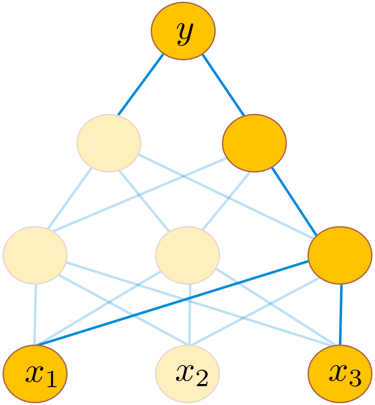
Dropout training



- Stochastic gradient descent
- Randomly remove every hidden/input node with prob. $\frac{1}{2}$ before each gradient descent update

[Hinton et al. 2012]

Dropout training



- Very successful in image recognition & speech recognition
- Why does it work?

  [Wagner, Wang, Liang 2013]
  [Helmbold, Long 2014]

What are we doing?

- Prove bounds for dropout
- Single neuron
- Linear loss

Outline

Online learning with expert

	E_1	E_2	E_3	$\dots$	E_n	$prediction$	label	loss
day 1	0	1	0	$\dots$	0	0	1	1

Online learning with expert

	E_1	E_2	E_3	$\ldots$	E_n	$prediction$	label	loss
day 1	0	1	0	$\ldots$	0	0	1	1
day 2	1	1	0	$\ldots$	0	1	1	0

Online learning with expert

	E_1	E_2	E_3	$\ldots$	E_n	*prediction*	label	loss
day 1	0	1	0	$\ldots$	0	0	1	1
day 2	1	1	0	$\ldots$	0	1	1	0
notation	x_1	x_1	x_2	$\ldots$	x_n	$\widehat{y}$	y	$\lvert\widehat{y} - y\rvert$

Online learning with expert

	E_1	E_2	E_3	$\dots$	E_n	*prediction*	label	loss
day 1	0	1	0	...	0	0	1	1
day 2	1	1	0	...	0	1	1	0
notation	x_1	x_1	x_2	$\dots$	x_n	$\widehat{y}$	y	$\lvert\widehat{y} - y\rvert$
scope	$\in [0,1]$			$\dots$		$\in [0,1]$	$\in \{0,1\}$	$\in [0,1]$

Online learning with expert

	E_1	E_2	E_3	$\ldots$	E_n	*prediction*	label	loss
day 1	0	1	0	...	0	0	1	1
day 2	1	1	0	...	0	1	1	0
notation	x_1	x_1	x_2	$\ldots$	x_n	$\widehat{y}$	y	$\lvert\widehat{y} - y\rvert$
scope	$\in [0, 1]$			$\ldots$		$\in [0, 1]$	$\in \{0, 1\}$	$\in [0, 1]$

- Algorithm maintains probability vector $\mathbf{w}$:
  - prediction $\widehat{y} = \mathbf{w} \cdot \mathbf{x}$

Online learning with expert

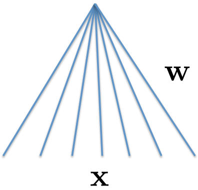
	E_1	E_2	E_3	$\dots$	E_n	*prediction*	label	loss
day 1	0	1	0	...	0	0	1	1
day 2	1	1	0	...	0	1	1	0
notation	x_1	x_1	x_2	$\dots$	x_n	$\widehat{y}$	y	$\lvert\widehat{y}-y\rvert$
scope	$\in [0,1]$			$\dots$		$\in [0,1]$	$\in \{0,1\}$	$\in [0,1]$

- Algorithm maintains probability vector $\mathbf{w}$:
  - prediction $\widehat{y} = \mathbf{w} \cdot \mathbf{x}$
- Loss linear because label $y \in \{0,1\}$
- $\underbrace{|\overbrace{\mathbf{w}\cdot\mathbf{x}}^{\widehat{y}} - y|}_{\text{loss of alg.}} = \sum_i \; w_i \; \underbrace{|x_i - y|}_{\text{loss of expert } i}$

Outline

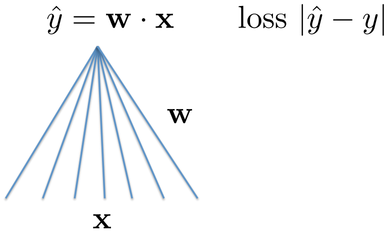
Predicting with expert advice

$\hat{y} = \mathbf{w} \cdot \mathbf{x}$ loss $|\hat{y} - y|$



On-line learning

Predicting with expert advice



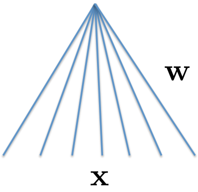
trial t

- get advice vector $\mathbf{x}_t$
- predict $\widehat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$
- get label y_t
- exp. losses $|x_{t,i} - y_t|$
- alg. loss $|\widehat{y}_t - y_t|$
- update $\mathbf{w}_t \to \mathbf{w}_{t+1}$

On-line learning

Predicting with expert advice

$\hat{y} = \mathbf{w} \cdot \mathbf{x}$ loss $|\hat{y} - y|$

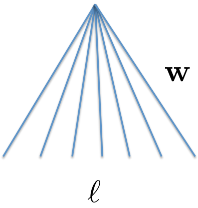


trial t

- get advice vector $\mathbf{x}_t$
- predict $\widehat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$
- get label y_t
- exp. losses $|x_{t,i} - y_t|$
- alg. loss $|\widehat{y}_t - y_t|$
- update $\mathbf{w}_t \to \mathbf{w}_{t+1}$

Hedge setting

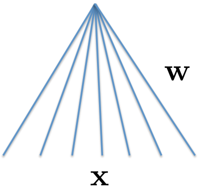
loss $\mathbf{w} \cdot \boldsymbol{\ell}$



On-line learning

Predicting with expert advice

$\hat{y} = \mathbf{w} \cdot \mathbf{x}$ loss $|\hat{y} - y|$

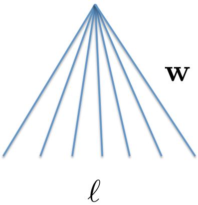


trial t

- get advice vector $\mathbf{x}_t$
- predict $\widehat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$
- get label y_t
- exp. losses $|x_{t,i} - y_t|$
- alg. loss $|\widehat{y}_t - y_t|$
- update $\mathbf{w}_t \to \mathbf{w}_{t+1}$

Hedge setting

loss $\mathbf{w} \cdot \ell$



trial t

-
- predict $\mathbf{w}_t$
- get loss vector $\boldsymbol{\ell}_t$
- exp. losses $\ell_{t,i}$
- alg. loss $\mathbf{w}_t \cdot \boldsymbol{\ell}_t$
- update $\mathbf{w}_t \to \mathbf{w}_{t+1}$

Predicting with a random expert

trial t

- predict $\mathbf{w}_t$ or predict with random expert $\widehat{i}_t$

Predicting with a random expert

trial t

- predict $\mathbf{w}_t$ — or predict with random expert $\widehat{i}_t$
- get loss vector $\boldsymbol{\ell}_t$
- alg. loss $\mathbf{w}_t \cdot \boldsymbol{\ell}_t$ — or alg. expected loss $\mathbb{E}\left[\ell_{\widehat{i}_t}\right] = \mathbf{w}_t \cdot \boldsymbol{\ell}_t$

Predicting with a random expert

trial t

- predict $\mathbf{w}_t$ — or predict with random expert $\widehat{i_t}$
- get loss vector $\boldsymbol{\ell}_t$
- alg. loss $\mathbf{w}_t \cdot \boldsymbol{\ell}_t$ — or alg. expected loss $\mathbb{E}\left[\ell_{\widehat{i_t}}\right] = \mathbf{w}_t \cdot \boldsymbol{\ell}_t$
- update $\mathbf{w}_t \rightarrow \mathbf{w}_{t+1}$

Predicting with a random expert

trial t

- predict $\mathbf{w}_t$ — or predict with random expert $\widehat{i}_t$
- get loss vector $\boldsymbol{\ell}_t$
- alg. loss $\mathbf{w}_t \cdot \boldsymbol{\ell}_t$ — or alg. expected loss $\mathbb{E}\left[\ell_{\widehat{i}_t}\right] = \mathbf{w}_t \cdot \boldsymbol{\ell}_t$
- update $\mathbf{w}_t \to \mathbf{w}_{t+1}$

weights are implicit

Only works for linear loss

How do we measure performance

Worst-case **regret**

$$\underbrace{\sum_{t=1}^{T} \mathbf{w}_t \cdot \boldsymbol{\ell}_t}_{\text{total expected loss of alg}} \quad - \quad \underbrace{\inf_i \ell_{\leq T,i}}_{\text{loss } \ell^* \text{ of best expert}}$$

Should be logarithmic in # of experts n

Outline

Main algorithms

	E_1	E_2	E_3	E_4	E_5
	0	1	0	0	1
	1	1	0	1	1
day $t-1$	0	0	1	1	1
$\ell_{\leq t-1,i}$					

Main algorithms

	E_1	E_2	E_3	E_4	E_5
	0	1	0	0	1
	1	1	0	1	1
day $t-1$	0	0	1	1	1
$\ell_{\leq t-1,i}$	1	2	1	2	3

Main algorithms

	E_1	E_2	E_3	E_4	E_5
	0	1	0	0	1
	1	1	0	1	1
day $t-1$	0	0	1	1	1
$\ell_{\leq t-1,i}$	1	2	1	2	3

FL $\qquad \widehat{i}_t = \mathrm{argmin}_i \ \ell_{\leq t-1,i}$ $\qquad$ ties broken uniformly

Main algorithms

	E_1	E_2	E_3	E_4	E_5
	0	1	0	0	1
	1	1	0	1	1
day $t-1$	0	0	1	1	1
$\ell_{\leq t-1,i}$	1	2	1	2	3

FL $\quad \widehat{i}_t = \mathrm{argmin}_i \; \ell_{\leq t-1,i}$ $\quad$ ties broken uniformly

FPL(η) $\quad \widehat{i}_t = \mathrm{argmin}_i \; \ell_{\leq t-1,i} + \frac{1}{\eta}\xi_{t,i}$ $\quad$ indep. additive noise

Main algorithms

	E_1	E_2	E_3	E_4	E_5
	0	1	0	0	1
	1	1	0	1	1
day $t-1$	0	0	1	1	1
$\ell_{\leq t-1,i}$	1	2	1	2	3

FL	$\widehat{i}_t = \operatorname{argmin}_i \ \ell_{\leq t-1,i}$	ties broken uniformly
FPL(η)	$\widehat{i}_t = \operatorname{argmin}_i \ \ell_{\leq t-1,i} + \frac{1}{\eta}\xi_{t,i}$	indep. additive noise
Hedge(η)	$w_i = \frac{e^{-\eta \ell_{t-1,i}}}{Z}$	Weighted Majority algorithm for pred. with Expert Advice Soft min

Dropout

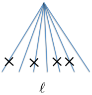
	E_1	E_2	E_3	E_4	E_5
	0	$\not{1}$	0	0	$\not{1}$
	1	1	0	1	1
day $t-1$	0	0	$\not{1}$	$\not{1}$	1
$\widehat{\ell}_{\leq t-1,i}$					

Dropout

	E_1	E_2	E_3	E_4	E_5
	0	$\not{1}$	0	0	$\not{1}$
	1	1	0	1	1
day $t-1$	0	0	$\not{1}$	$\not{1}$	1
$\widehat{\ell}_{\leq t-1,i}$	1	1	0	1	2

$$\widehat{\ell}_{t,i} = \begin{cases} 0 & \text{with prob. } \alpha \\ \ell_{t,i} & \text{otherwise} \end{cases}$$

indep. multiplicative noise



FL on dropout $\quad \widehat{i}_t = \operatorname{argmin}_i \widehat{\ell}_{\leq t-1,i}$

How good?

Optimal worst case regret: $\sqrt{L^* \ln n} + \ln n$

How good?

Optimal worst case regret: $\sqrt{L^* \ln n} + \ln n$

- FL is bad
- FPL(η) and Hedge(η) achieve optimal regret with tuning
  - fancy tunings: AdaHedge and Flipflop

How good?

Optimal worst case regret: $\sqrt{L^* \ln n} + \ln n$

- FL is bad
- FPL(η) and Hedge(η) achieve optimal regret with tuning
  - fancy tunings: AdaHedge and Flipflop
- FL on dropout requires no tuning

How good?

Optimal worst case regret: $\sqrt{L^* \ln n} + \ln n$

- FL is bad
- FPL(η) and Hedge(η) achieve optimal regret with tuning
  - fancy tunings: AdaHedge and Flipflop
- FL on dropout requires no tuning
  - dropout better noise for achieving optimal worst case regret
  - in iid case with gap between 1st and 2nd: $\log n$ regret

What regularization?

Hedge(η)	relative entropy

What regularization?

Hedge(η)	relative entropy
FPL(η)	additive $\frac{1}{\eta}$ log exponential noise $=$ Hedge(η)

What regularization?

Hedge(η)	relative entropy
FPL(η)	additive $\frac{1}{\eta}$ log exponential noise $=$ Hedge(η)
FL on dropout	tricky

Feed forward NN	[Wagner, Wang, Liang 2013]
Logistic regression	[Helmbold, Long 2014]
Linear loss case	[ALST 2014]

Our path to dropout

- Loss vectors $\boldsymbol{\ell}_t \quad \longrightarrow \quad$ loss matrices $\mathbf{L}_t$
- Prob. vectors $\mathbf{w}_t \quad \longrightarrow \quad$ density matrices $\mathbf{W}_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq \boldsymbol{\ell} \leq t-1,i}}}{Z} \quad \longrightarrow \quad$ Matrix Hedge $\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1,i}\right)}{Z'}$
- Matrix Hedge $O(n^3)$ per update

Our path to dropout

- Loss vectors $\boldsymbol{\ell}_t \quad \longrightarrow \quad$ loss matrices $\mathbf{L}_t$
- Prob. vectors $\mathbf{w}_t \quad \longrightarrow \quad$ density matrices $\mathbf{W}_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq \boldsymbol{\ell} \leq t-1,i}}}{Z} \quad \longrightarrow \quad$ Matrix Hedge
  $\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1,i}\right)}{Z'}$
- Matrix Hedge $O(n^3)$ per update
- FL minimum eigenvector calculation of $\mathbf{L}_{\leq t-1,i}$: $\quad O(n^2)$

Our path to dropout

- Loss vectors $\boldsymbol{\ell}_t \quad \longrightarrow \quad$ loss matrices $\mathbf{L}_t$
- Prob. vectors $\mathbf{w}_t \quad \longrightarrow \quad$ density matrices $\mathbf{W}_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq \boldsymbol{\ell} \leq t-1,i}}}{Z} \quad \longrightarrow \quad$ Matrix Hedge $\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1,i}\right)}{Z'}$
- Matrix Hedge $O(n^3)$ per update

- FL minimum eigenvector calculation of $\mathbf{L}_{\leq t-1,i}$: $O(n^2)$
- Is there $O(n^2)$ perturbation with optimal regret bound?

Our path to dropout

- Loss vectors $\boldsymbol{\ell}_t \quad \longrightarrow \quad$ loss matrices $\mathbf{L}_t$
- Prob. vectors $\mathbf{w}_t \quad \longrightarrow \quad$ density matrices $\mathbf{W}_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq \boldsymbol{\ell} \leq t-1,i}}}{Z} \quad \longrightarrow \quad$ Matrix Hedge $\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1,i}\right)}{Z'}$
- Matrix Hedge $O(n^3)$ per update

- FL minimum eigenvector calculation of $\mathbf{L}_{\leq t-1,i}$: $\quad O(n^2)$
- Is there $O(n^2)$ perturbation with optimal regret bound?

- Follow the skipping leader: Drop entire loss $\mathbf{L}_t$ with probability $\frac{1}{2}$

Our path to dropout

- Loss vectors $\boldsymbol{\ell}_t \quad \longrightarrow \quad$ loss matrices $\mathbf{L}_t$
- Prob. vectors $\mathbf{w}_t \quad \longrightarrow \quad$ density matrices $\mathbf{W}_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq \boldsymbol{\ell} \leq t-1,i}}}{Z} \quad \longrightarrow \quad$ Matrix Hedge $\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1,i}\right)}{Z'}$
- Matrix Hedge $O(n^3)$ per update

- FL minimum eigenvector calculation of $\mathbf{L}_{\leq t-1,i}$: $O(n^2)$
- Is there $O(n^2)$ perturbation with optimal regret bound?

- Follow the skipping leader: Drop entire loss $\mathbf{L}_t$ with probability $\frac{1}{2}$
- Proof techniques break down - settled for vector case and independent multiplicative noise = dropout

Our path to dropout

- Loss vectors $\boldsymbol{\ell}_t \quad \longrightarrow \quad$ loss matrices $\mathbf{L}_t$
- Prob. vectors $\mathbf{w}_t \quad \longrightarrow \quad$ density matrices $\mathbf{W}_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq \boldsymbol{\ell} \leq t-1,i}}}{Z} \quad \longrightarrow \quad$ Matrix Hedge $\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1,i}\right)}{Z'}$
- Matrix Hedge $O(n^3)$ per update

- FL minimum eigenvector calculation of $\mathbf{L}_{\leq t-1,i}$: $O(n^2)$
- Is there $O(n^2)$ perturbation with optimal regret bound?

- Follow the skipping leader: Drop entire loss $\mathbf{L}_t$ with probability $\frac{1}{2}$
- Proof techniques break down - settled for vector case and independent multiplicative noise = dropout
- ~~Follow the skipping leader~~ can have linear regret

[Lugosi, Neu 2014]

Outline

Simple algorithms

Any deterministic alg. (such as FL) has huge regret

- For T trials: give algorithm's expert a unit of loss
- Loss of alg.: T loss of best: $\leq \frac{T}{n}$

Simple algorithms

Any deterministic alg. (such as FL) has huge regret

- For T trials: give algorithm's expert a unit of loss
- Loss of alg.: T loss of best: $\leq \frac{T}{n}$

regret: $\geq \underbrace{T}_{nL^*} - \underbrace{\frac{T}{n}}_{L^*} = (n-1)L^*$

Simple algorithms

Any deterministic alg. (such as FL) has huge regret

- For T trials: give algorithm's expert a unit of loss
- Loss of alg.: T loss of best: $\leq \frac{T}{n}$

regret: $\geq \underbrace{T}_{nL^*} - \underbrace{\frac{T}{n}}_{L^*} = (n-1)L^*$

Recall optimum regret: $\sqrt{L^* \ln n} + \ln n$

FL with random ties

Simple algorithms

Any deterministic alg. (such as FL) has huge regret

- For T trials: give algorithm's expert a unit of loss
- Loss of alg.: T loss of best: $\leq \frac{T}{n}$

  regret: $\geq \underbrace{T}_{nL^*} - \underbrace{\frac{T}{n}}_{L^*} = (n-1)L^*$

Recall optimum regret: $\sqrt{L^* \ln n} + \ln n$

FL with random ties

- Give every expert one unit of loss
  - iterate $L^* + 1$ times
- Loss per sweep $\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{2} + 1 \approx \ln n$
- Loss of alg.: $(L^* + 1) \ln n$ loss of best: L^*

  regret: $L^* \ln n$

Analysis of dropout

Unit rule

- Adversary forces more regret by splitting loss vectors into units

$$\begin{pmatrix} \mathbf{1} \\ 0 \\ \mathbf{1} \\ \mathbf{1} \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{1} \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \mathbf{1} \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathbf{1} \end{pmatrix}$$

Analysis of dropout

Unit rule

- Adversary forces more regret by splitting loss vectors into units

$$\begin{pmatrix}1\\0\\1\\1\end{pmatrix} \longrightarrow \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

Swapping rule

E_1	1	1	1	1	1	1	1	1	1		9
E_2	1	1	1	1	1	1	1	1			8
E_3	1	1	1	1	1	1	1	1	1	1	10
E_4	1	1	1	1	1	1					6

Analysis of dropout

Unit rule

- Adversary forces more regret by splitting loss vectors into units

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Swapping rule

E_1	1	1	1	1	1	1	**1**	1	1		9
E_2	1	1	1	1	1	1	1	1			8
E_3	1	1	1	**1**	1	1	1	1	1	1	10
E_4	1	1	1	1	1	1					6

- 1's in some order
- **1** before **1**
- Otherwise adversary benefits from swapping

Worst-case pattern

1	1	1	1	1				
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

Cost per sweep

Assume we have s leaders

Cost per sweep

Assume we have s leaders

$$s \text{ leader get unit} \atop \text{ignore non-leaders} \quad \left\{ \begin{array}{l} 1 \\ \;1 \\ \;\;1 \\ \;\;\;1 \\ \;\;\;\;1 \end{array} \right.$$

Cost per sweep

Assume we have s leaders

$$s \text{ leader get unit} \quad \text{ignore non-leaders} \quad \left\{ \begin{array}{l} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \right.$$

FL

$$\frac{1}{s} + \frac{1}{s-1} + \frac{1}{s-2} + \frac{1}{s-3} + \ldots + \frac{1}{\underbrace{s-s-2}_{2}} + \frac{1}{\underbrace{s-s-1}_{1}}$$

$$\approx \ln s$$

Cost per sweep

Assume we have s leaders

s leader get unit
ignore non-leaders

$$\begin{cases} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{cases}$$

FL

$$\frac{1}{s} + \frac{1}{s-1} + \frac{1}{s-2} + \frac{1}{s-3} + \ldots + \frac{1}{\underbrace{s-s-2}_{2}} + \frac{1}{\underbrace{s-s-1}_{1}}$$

$$\approx \ln s$$

Dropout

$$\frac{1}{s} + \frac{1}{s-1/2} + \frac{1}{s-2/2} + \frac{1}{s-3/2} + \ldots + \frac{1}{s-(s-2)/2} + \frac{1}{s-(s-1)/2}$$

$$\approx 2\ln\frac{2s}{s} = 2\ln 2$$

$L^* = 0$ - one expert incurs no loss

FL

- One sweep

$$\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{2} \cancel{+1} \approx (\ln n) - 1$$

- Optimal

$L^* = 0$ - one expert incurs no loss

FL

- One sweep

$$\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{2} \cancel{+1} \approx (\ln n) - 1$$

- Optimal

Dropout

- # of leaders reduced by half in each sweep
- $\approx \log_2 n$ sweeps times $\leq 2 \ln 2 = 1.386$

$$=========================$$

$2 \ln n$

Overview of proof for noisy case

- Focus on first L sweeps
- Only occurs constant regret if number of leaders > 1

Overview of proof for noisy case

- Focus on first L sweeps
- Only occurs constant regret if number of leaders > 1
- Prob. that number of leaders > 1 is at most $\sqrt{\frac{\ln n}{q+1}}$ for sweep q

Overview of proof for noisy case

- Focus on first L sweeps
- Only occurs constant regret if number of leaders > 1
- Prob. that number of leaders > 1 is at most $\sqrt{\frac{\ln n}{q+1}}$ for sweep q
- For Hedge(η) and FPL(η) cost per sweep constant and dependent on η

Outlook

- Combinatorial experts
- Matrix case
- Where else can dropout perturbations be used?
- Dropout for convex losses
- Dropout for neural nets