Adversarial Attacks on Probabilistic Autoregressive Forecasting Models

ICML 2020


Raphaël Dang-Nhu


Gagandeep Singh

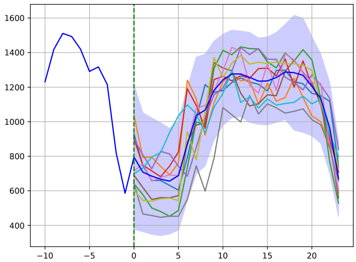

Pavol Bielik


Martin Vechev

Department of Computer Science, ETH Zürich

dangnhur@ethz.ch

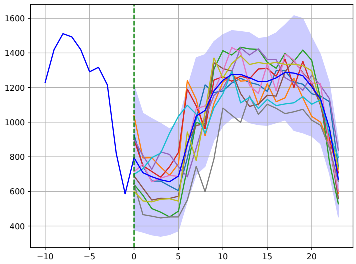
Neural architectures with stochastic behavior



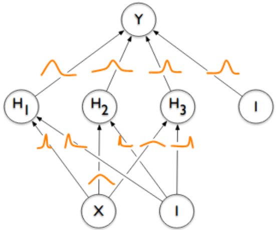
(i) Probabilistic forecasting model

[1] Blundell et al., Weight Uncertainty in Neural Networks, ICML 2015

Neural architectures with stochastic behavior



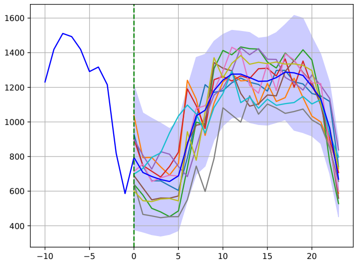
(i) Probabilistic forecasting model



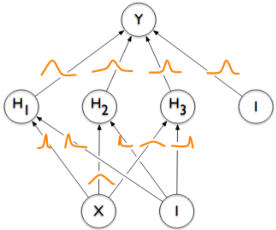
(ii) Bayesian neural network

[1]Blundell et al., Weight Uncertainty in Neural Networks, ICML 2015

Neural architectures with stochastic behavior



(i) Probabilistic forecasting model

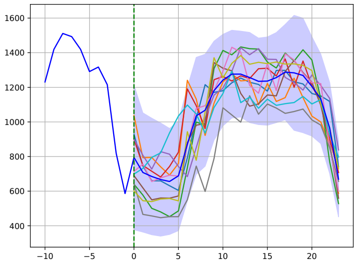


(ii) Bayesian neural network

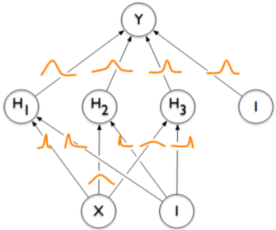
- Multiple sources of noise: (i) each timestep, (ii) each weight[1]

[1] Blundell et al., Weight Uncertainty in Neural Networks, ICML 2015

Neural architectures with stochastic behavior



(i) Probabilistic forecasting model



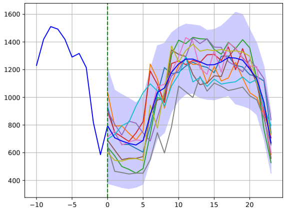
(ii) Bayesian neural network

- Multiple sources of noise: (i) each timestep, (ii) each weight[1]
- Complex resulting output distribution, approximated via Monte-Carlo sampling

[1]Blundell et al., Weight Uncertainty in Neural Networks, ICML 2015

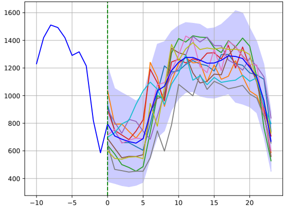
Focus of this work: probabilistic forecasting models

- Stochastic sequence model
- Generates several prediction traces

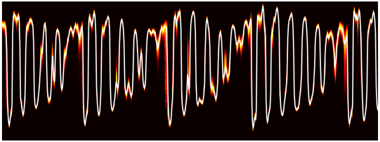


Focus of this work: probabilistic forecasting models

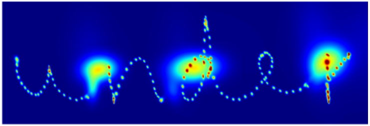
- Stochastic sequence model
- Generates several prediction traces



Traditionally used as a generative model



WaveNet for raw audio



Handwriting generation

Probabilistic forecasting models for decision-making[2]

- Allows to predict volatility of the time-series.
- Useful with low signal-to-noise ratio.

Key idea: use generated traces as Monte-Carlo samples to estimate the evolution of the time-series

[2]Salinas et al., DeepAR: Probabilistic forecasting with autoregressive recurrent networks, International Journal of Forecasting, 2020

Probabilistic forecasting models for decision-making[2]

- Allows to predict volatility of the time-series.
- Useful with low signal-to-noise ratio.

Key idea: use generated traces as Monte-Carlo samples to estimate the evolution of the time-series

Stock prices

Electricity consumption

Business sales

Integrated in **Amazon Sagemaker** (DeepAR architecture)

[2]Salinas et al., DeepAR: Probabilistic forecasting with autoregressive recurrent networks, International Journal of Forecasting, 2020

Contributions

- New class of attack objectives based on output **statistics**

Contributions

- New class of attack objectives based on output **statistics**
- Adaptation of gradient-based adversarial attacks to these new attack objectives for stochastic models

Contributions

- New class of attack objectives based on output **statistics**
- Adaptation of gradient-based adversarial attacks to these new attack objectives for stochastic models
- Main technical aspect: developing estimators for propagating the objective gradient through the Monte-Carlo approximation

Contributions

- New class of attack objectives based on output **statistics**
- Adaptation of gradient-based adversarial attacks to these new attack objectives for stochastic models
- Main technical aspect: developing estimators for propagating the objective gradient through the Monte-Carlo approximation

Contributions

- New class of attack objectives based on output **statistics**
- Adaptation of gradient-based adversarial attacks to these new attack objectives for stochastic models
- Main technical aspect: developing estimators for propagating the objective gradient through the Monte-Carlo approximation

We aim at providing an **off-the-shelf methodology** for these attacks

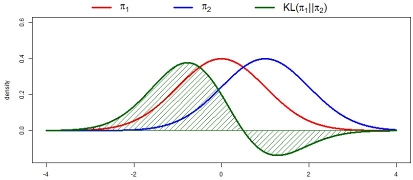
Class of attack objectives

Stochastic model with input x, and output $y \sim q_x(\cdot)$.
Previously considered attack objectives:

Stochastic model with input x, and output $y \sim q_x(\cdot)$. Previously considered attack objectives:

Untargeted attacks on information divergence D with the original predicted distribution

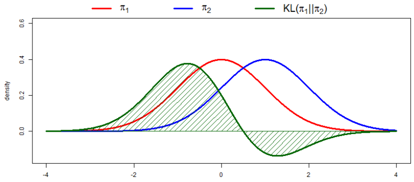
$$\max_{\delta} D\left(q_{x+\delta} \| q_x\right)$$

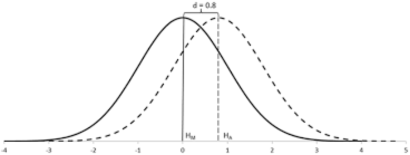


Stochastic model with input x, and output $y \sim q_x(\cdot)$.
Previously considered attack objectives:

Untargeted attacks on information divergence D with the original predicted distribution

$$\max_{\delta} D\left(q_{x+\delta} \| q_x\right)$$





Untargeted/Targeted attacks on the mean of the distribution

$$\min_{\delta} \text{distance}\left(\mathbb{E}_{q_{x+\delta}}[y], \text{target}\right)$$

Framework

We perform a targeted attack on a **statistic** $\chi(y)$ of the output.

Framework

We perform a targeted attack on a **statistic** $\chi(y)$ of the output.
This corresponds to minimizing

$$\text{distance}\left(\mathbb{E}_{q_{x+\delta}}[\chi(y)], \text{target}\right)$$

Framework

We perform a targeted attack on a **statistic** $\chi(y)$ of the output. This corresponds to minimizing

$$\mathrm{distance}\left(\mathbb{E}_{q_{x+\delta}}[\chi(y)], \mathrm{target}\right)$$

Extensions:

- Bayesian setting $q_x(y|z)$.
- Generalization to simultaneous attack of several statistics.
- Statistics depending on x.

Motivation 1: option pricing in finance

Consider a stock with

- past prices $x = (p_1, \ldots, p_{t-1})$
- predicted future prices $y = (p_t, \ldots, p_T)$.

Motivation 1: option pricing in finance

Consider a stock with

- past prices $x = (p_1, \ldots, p_{t-1})$
- predicted future prices $y = (p_t, \ldots, p_T)$.

Name	$\chi(y)$	Observation z
European call option	$\max(0, y_h)$	
Asian call option	$\mathrm{average}_i(y_i)$	
Limit sell order	$\mathbb{1}\left[\max_i y_i \geq \mathrm{threshold}\right]$	
Barrier option	y_h	$\max_i y_i \geq \mathrm{threshold}$

Motivation 1: option pricing in finance

Consider a stock with

- past prices $x = (p_1, \ldots, p_{t-1})$
- predicted future prices $y = (p_t, \ldots, p_T)$.

Name	$\chi(y)$	Observation z
European call option	$\max(0, y_h)$	
Asian call option	$\mathrm{average}_i(y_i)$	
Limit sell order	$\mathbb{1}\left[\max_i y_i \geq \mathrm{threshold}\right]$	
Barrier option	y_h	$\max_i y_i \geq \mathrm{threshold}$

Our framework allows to specifically target one of these options

Motivation 2: attacking model uncertainty

Some defenses use **prediction uncertainty** to detect adversarial examples.

Motivation 2: attacking model uncertainty

Some defenses use **prediction uncertainty** to detect adversarial examples.

New attacks bypass these defenses by enforcing **uncertainty constraints** for the adversarial example.

Motivation 2: attacking model uncertainty

Some defenses use **prediction uncertainty** to detect adversarial examples.

New attacks bypass these defenses by enforcing **uncertainty constraints** for the adversarial example.

Our framework allows to express these constraints, with

- The entropy $\mathbb{E}_{q_x}[-\log(q[y|x])]$.
- The distribution's moments $\mathbb{E}_{q_x}[y^k]$.

Details about the estimators

Technical challenge

Gradient-based attacks require computing

$$\nabla_{\boldsymbol{\delta}} \mathbb{E}_{q[\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\delta},z]}[\chi(\boldsymbol{y})]$$

Technical challenge

Gradient-based attacks require computing

$$\nabla_{\boldsymbol{\delta}} \mathbb{E}_{q[\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\delta},z]}[\chi(\boldsymbol{y})]$$

The expectation and its gradient have no analytical closed form

Technical challenge

Gradient-based attacks require computing

$$\nabla_{\boldsymbol{\delta}} \mathbb{E}_{q[\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\delta},z]}[\chi(\boldsymbol{y})]$$

The expectation and its gradient have no analytical closed form

We provide two different estimators to approximate the gradient

Approach 1: REINFORCE

- A.k.a as log-derivative trick and score-function estimator.
- Based on interversion of expectation and derivative.

Approach 1: REINFORCE

- A.k.a as log-derivative trick and score-function estimator.
- Based on interversion of expectation and derivative.

$$\nabla_{\boldsymbol{\delta}} \mathbb{E}_{q[\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\delta},z]}[\chi(\boldsymbol{y})] \simeq \frac{\sum_{l=1}^{L} \chi(\boldsymbol{y}^l) q[z|\boldsymbol{x}+\boldsymbol{\delta},\boldsymbol{y}^l] \nabla_{\boldsymbol{\delta}} \log(q[\boldsymbol{y}^l|\boldsymbol{x}+\boldsymbol{\delta},z])}{\sum_{l=1}^{L} q[z|\boldsymbol{x}+\boldsymbol{\delta},\boldsymbol{y}^l]}$$

REINFORCE estimator

.

Approach 2: Reparametrization

- Mitigates the high-variance of REINFORCE.
- Typically used for variational inference.
- Assumes a reparametrization $y \sim g(x, \eta)$, where g is deterministic.

Approach 2: Reparametrization

- Mitigates the high-variance of REINFORCE.
- Typically used for variational inference.
- Assumes a reparametrization $y \sim g(x, \eta)$, where g is deterministic.

$$\nabla_{\boldsymbol{\delta}} \mathbb{E}_{q[\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\delta},z]}[\chi(\boldsymbol{y})] \simeq \nabla_{\boldsymbol{\delta}} \left(\frac{\sum_{l=1}^{L} \chi(g_{\boldsymbol{x}}(\boldsymbol{\delta}, \boldsymbol{\eta}^l)) q[z|\boldsymbol{x}+\boldsymbol{\delta}, g_{\boldsymbol{x}}(\boldsymbol{\delta}, \boldsymbol{\eta}^l)]}{\sum_{l=1}^{L} q[z|\boldsymbol{x}+\boldsymbol{\delta}, g_{\boldsymbol{x}}(\boldsymbol{\delta}, \boldsymbol{\eta}^l)]} \right)$$

Reparametrization estimator

.

Comparison

Respective advantages of gradient estimators.

Method	REINFORCE	Reparametrization
Applies to non-differentiable statistics	✔	
Requires no reparametrization	✔	
Applies to Bayesian setting		✔
Yields best gradient estimates		✔

Comparison

Respective advantages of gradient estimators.

Method	REINFORCE	Reparametrization
Applies to non-differentiable statistics	✔	
Requires no reparametrization	✔	
Applies to Bayesian setting		✔
Yields best gradient estimates		✔

Detailed comparison and conditions in the paper!

Experimental evaluation

Experiments: stock prices

Algorithmic trading scenario, standard additive threat model, maximum Euclidean norm of 0.1[3] for the perturbation.

[3]Corresponds to perturbing one value by 10%, 10 values by 3.3%, 100 values by 1%.

Experiments: stock prices

Algorithmic trading scenario, standard additive threat model, maximum Euclidean norm of 0.1[3] for the perturbation.

- Attack is successful on 90% of test inputs.

[3]Corresponds to perturbing one value by 10%, 10 values by 3.3%, 100 values by 1%.

Experiments: stock prices

Algorithmic trading scenario, standard additive threat model, maximum Euclidean norm of 0.1[3] for the perturbation.

- Attack is successful on 90% of test inputs.
- The network incurs a daily financial loss of −13%.

[3] Corresponds to perturbing one value by 10%, 10 values by 3.3%, 100 values by 1%.

Experiments: stock prices

Algorithmic trading scenario, standard additive threat model, maximum Euclidean norm of 0.1[3] for the perturbation.

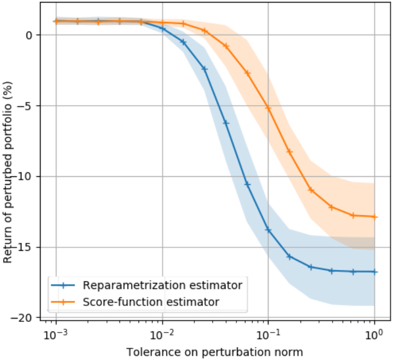
- Attack is successful on 90% of test inputs.
- The network incurs a daily financial loss of −13%.

[3] Corresponds to perturbing one value by 10%, 10 values by 3.3%, 100 values by 1%.

Experiments: stock prices

Algorithmic trading scenario, standard additive threat model, maximum Euclidean norm of 0.1[3] for the perturbation.

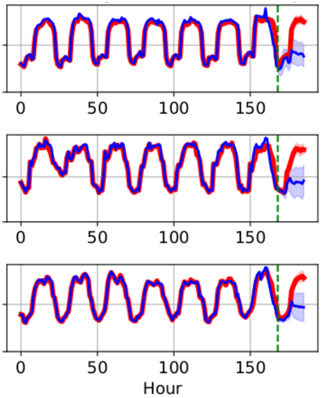
- Attack is successful on 90% of test inputs.
- The network incurs a daily financial loss of −13%.



[3]Corresponds to perturbing one value by 10%, 10 values by 3.3%, 100 values by 1%.

Experiments: electricity

Original test samples (red) and adversarial examples (blue) for prediction of electricity consumption.



Thanks for listening

Code and trained models are available at

`github.com/eth-sri/probabilistic-forecasts-attacks`

Contact at

dangnhur@ethz.ch