Cuantificación de la incertidumbre en la predicción espacio-temporal con bayesian deep learning: Aplicación a El Niño

Autor: Jose González Abad
Director: Jorge Baño Medina

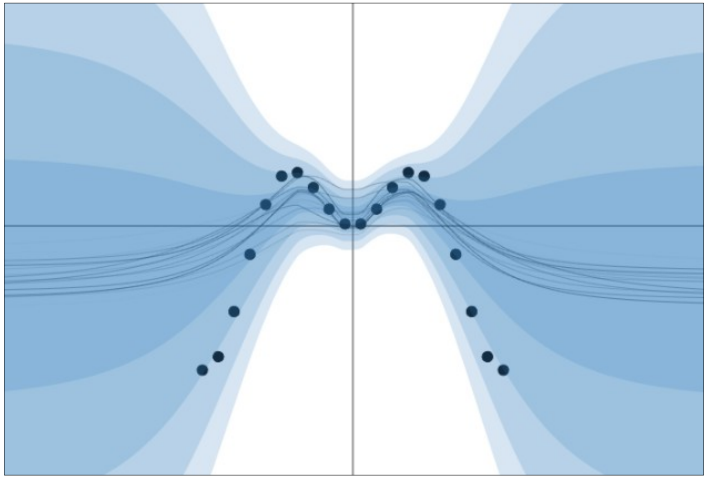
ÍNDICE

Introducción

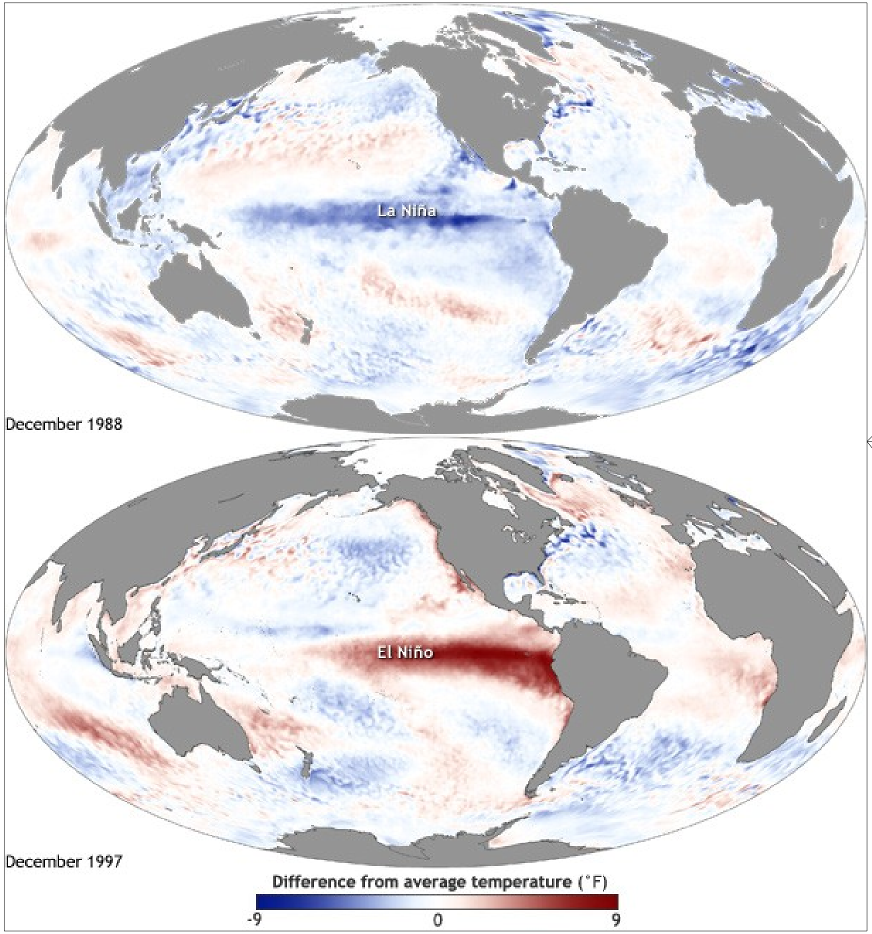
INTRODUCCIÓN

El Niño

Cuantificación de la incertidumbre



El Niño

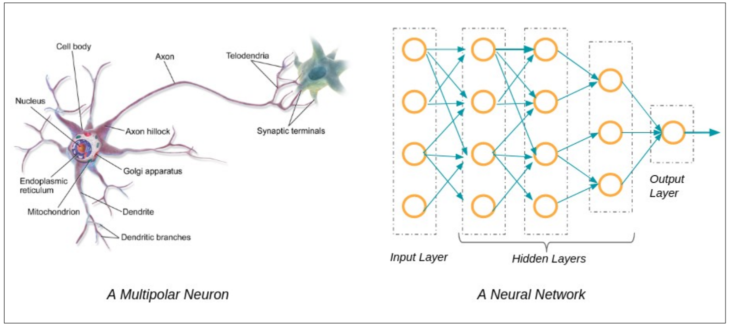


La Niña
December 1988
El Niño
December 1997
Difference from average temperature (˚F)
-9
0
9

FLOODED

El Niño

Modelos físico-matemáticos



Lineales

Machine Learning

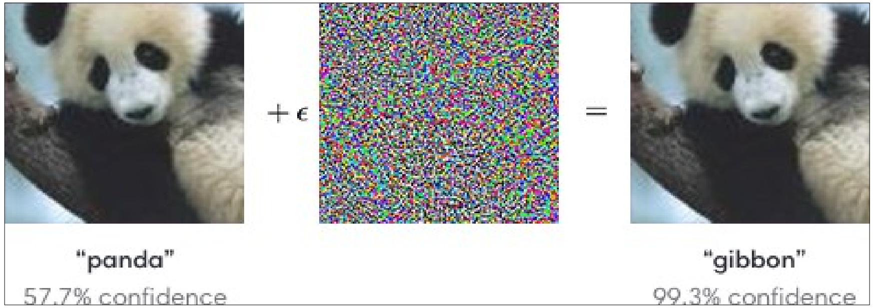
Modelos estadísticos

No lineales

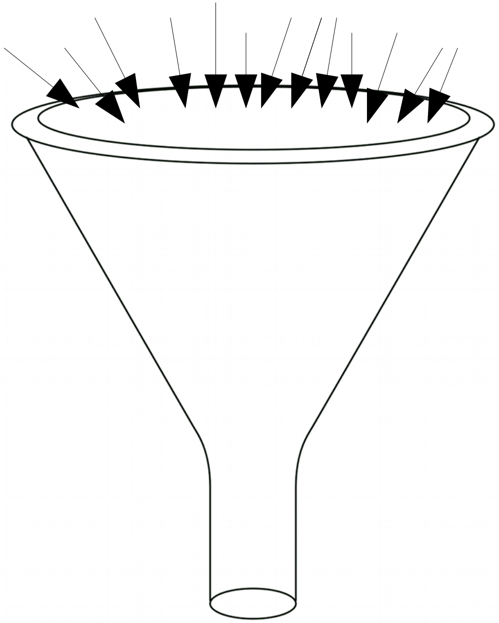
Deep Learning

Cuantificación de la incertidumbre

¿Cómo de seguros están nuestros modelos?



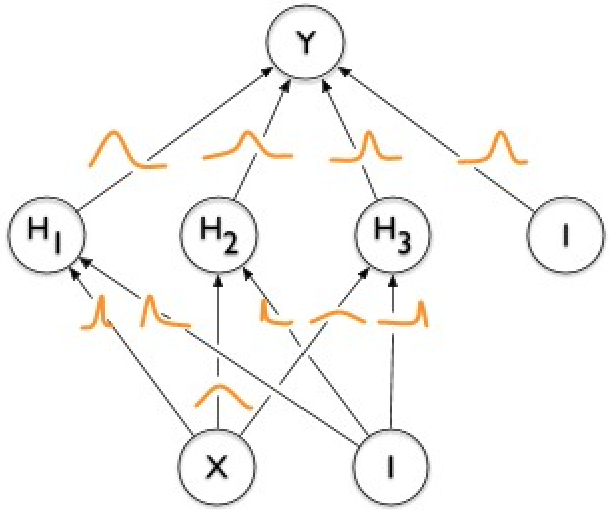
Intervalo de confianza



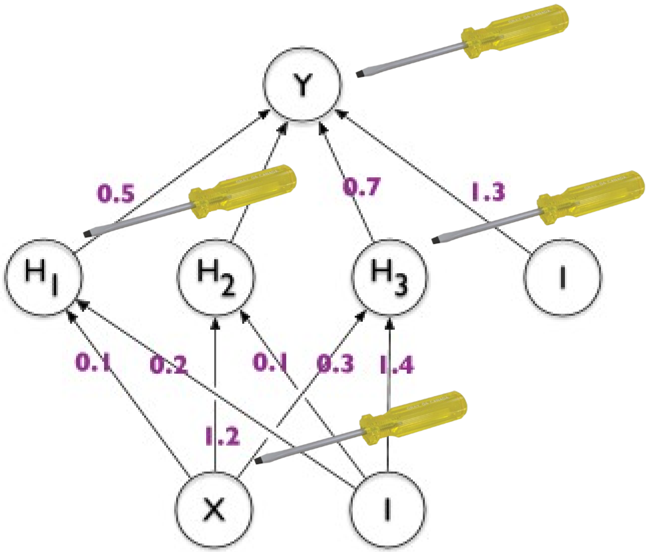
0.9

Cuantificación de la incertidumbre

Redes neuronales bayesianas



Bayesian deep learning



Predicción de El Niño

Bayesian Recurrent Neural Network Models for Forecasting and Quantifying Uncertainty in Spatial-Temporal Data

Patrick L. McDermott* Christopher K. Wikle

Department of Statistics

University of Missouri

February 8, 2018

Abstract

Recurrent neural networks (RNNs) are nonlinear dynamical models commonly used in the machine learning and dynamical systems literature to represent complex dynamical or sequential relationships between variables. More recently, as deep learning models have become more common, RNNs have been used to forecast increasingly complicated systems. Dynamical spatio-temporal processes represent a class of complex systems that can potentially benefit from these types of models. Although the RNN literature is expansive and highly developed, uncertainty quantification is often ignored. Even when considered, the uncertainty is generally quantified without the use of a rigorous framework, such as a fully Bayesian setting. Here we attempt to quantify uncertainty in a more formal framework while maintaining the forecast accuracy that

*Department of Statistics, University of Missouri, 146 Middlebush Hall, Columbia, MO 65211 USA; E-mail: plmyt7@mail.missouri.edu (corresponding author)

vs

DEEP LEARNING

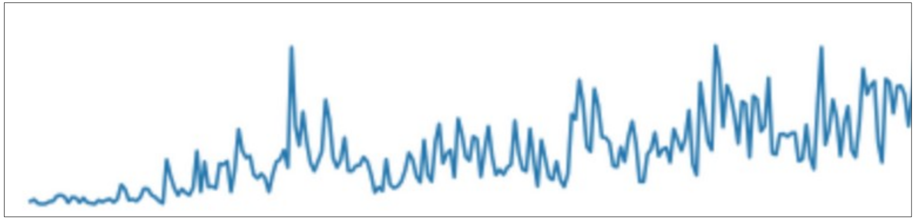
+

Bayesian deep learning

Deep Learning aplicado al Forecasting

Deep Learning aplicado al Forecasting

El Niño presenta una marcada componente temporal



Enfoque determinista
Metodología Box-Jenkins

Univariante

VAR

Multivariante

Lineal

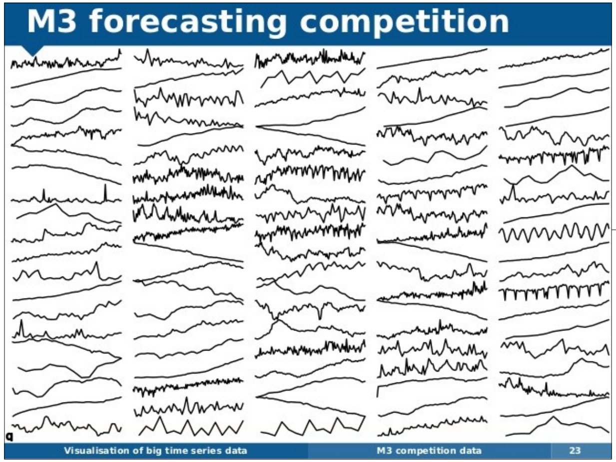
Con intervalos de confianza

Machine Learning
Redes Neuronales

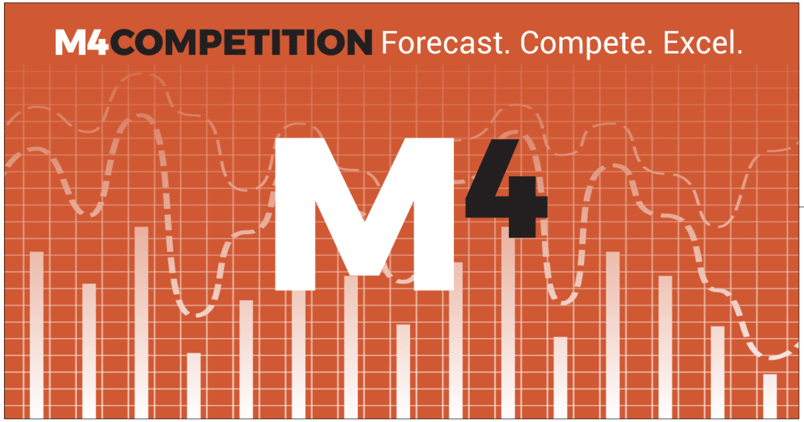
No lineal

Sin intervalos de confianza

Deep Learning aplicado al Forecasting

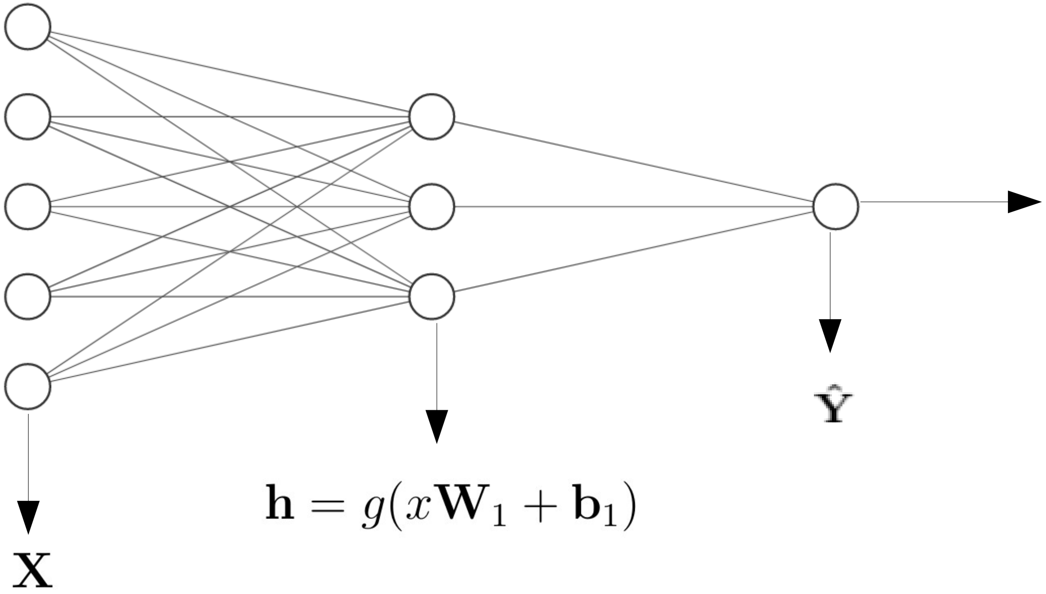


Método Theta



Red neuronal recurrente + suavizados exponenciales

Red neuronal densa



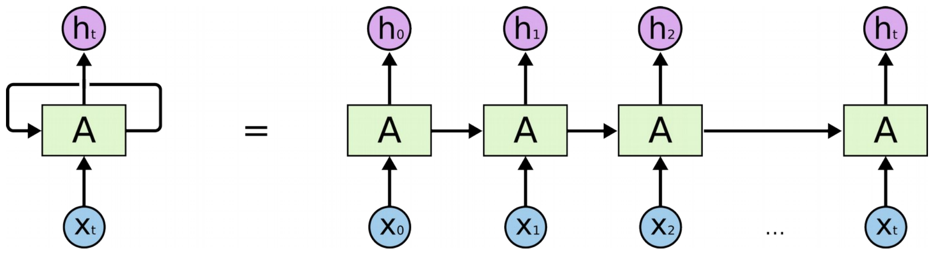
Minimizar diferencias entre las predicción y valor real

Backpropagation

Gradient descent

Red neuronal recurrente

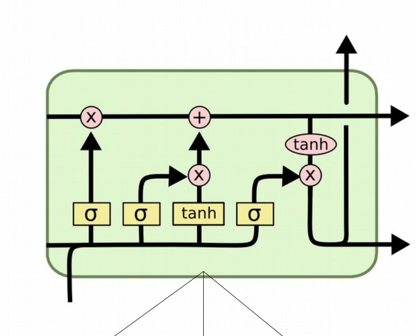
Red neuronal recurrente



$$\mathbf{h}_t = g(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1})$$

Vanishing Gradient

LSTM

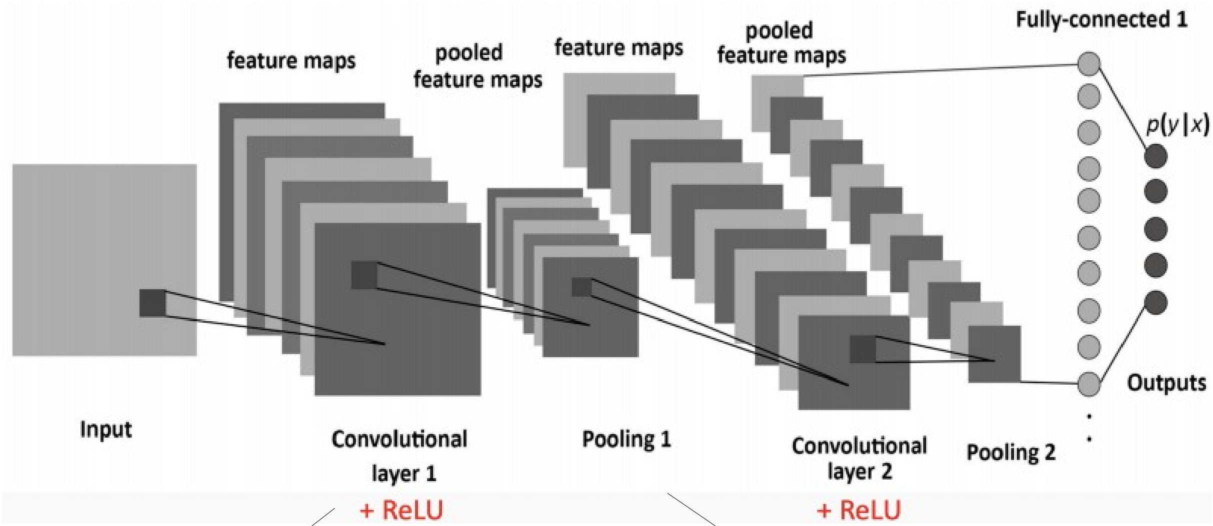


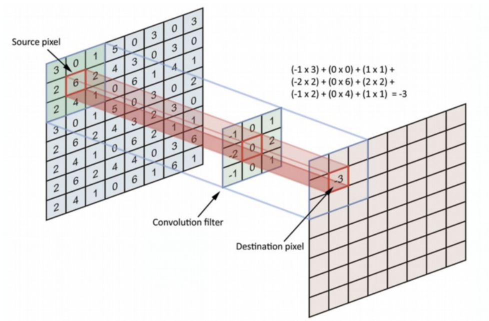
$$f_t = \sigma(\mathbf{W}_j[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f)$$

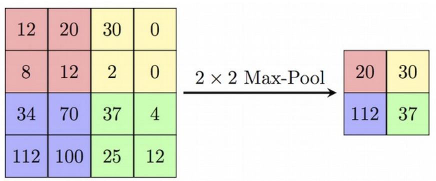
$$\mathbf{i}_t := \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i)$$
$$\mathbf{c}'_t := tanh(\mathbf{W}_C[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C)$$

$$\mathbf{o}_t := \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o)$$
$$\mathbf{h}_t := \mathbf{o}_t tanh(\mathbf{c}_t)$$

Red neuronal convolucional





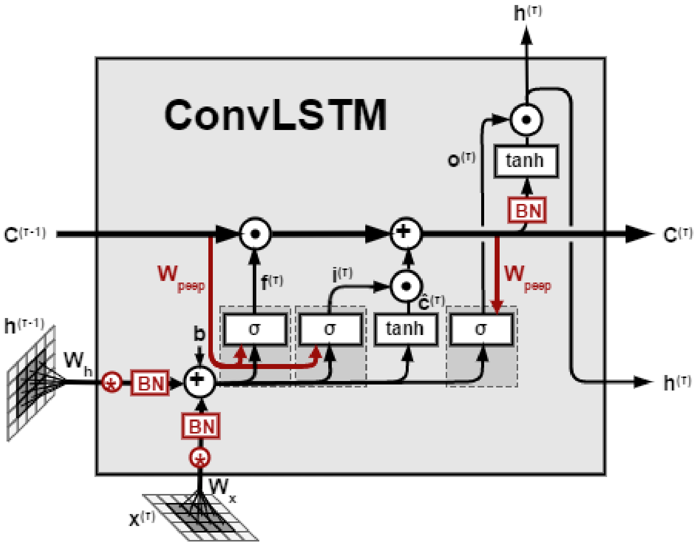


Red neuronal convolucional

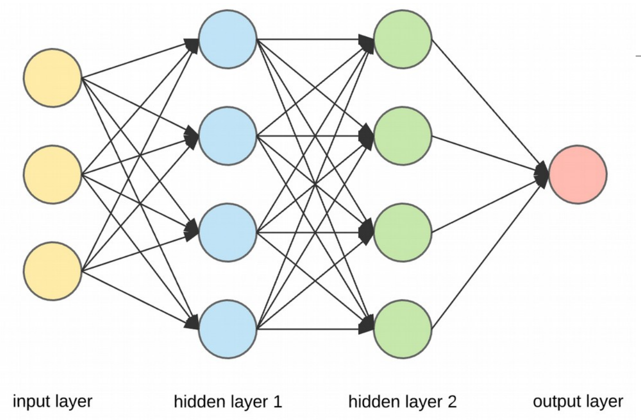
Dinámica temporal → LSTM

Dinámica espacial → Convolucional

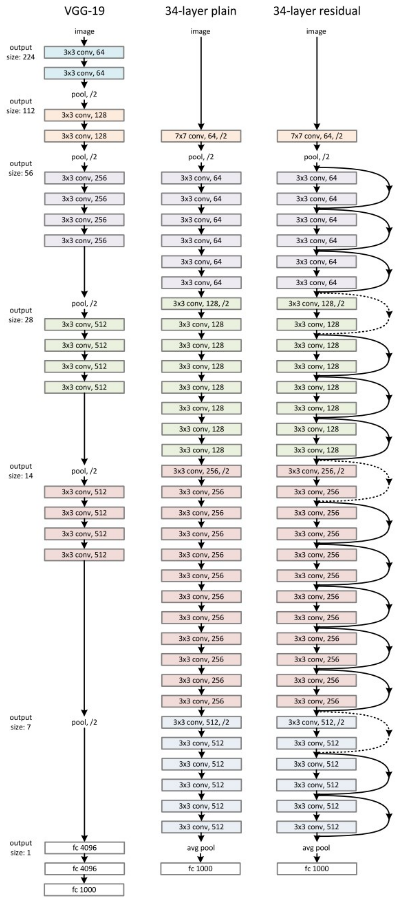
LSTM Convolucional



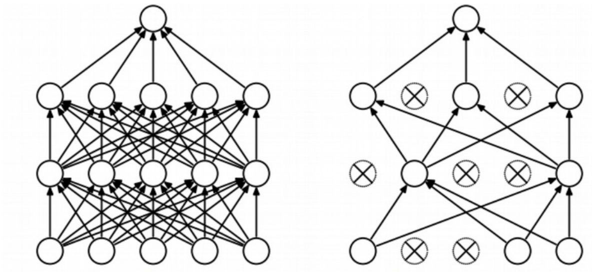
Deep learning



- Incremento exponencial en la cantidad de datos
- Desarrollo de hardware específico
- Desarrollo de nuevos algoritmos



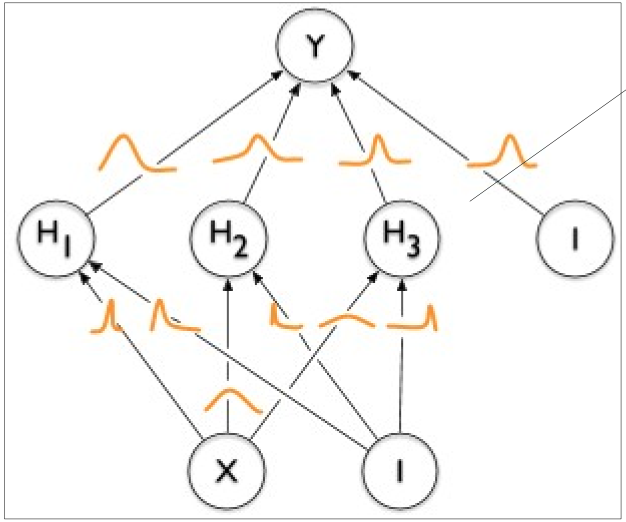
Dropout



(a) Standard Neural Net (b) After applying dropout.

Bayesian deep learning

Red neuronal bayesiana



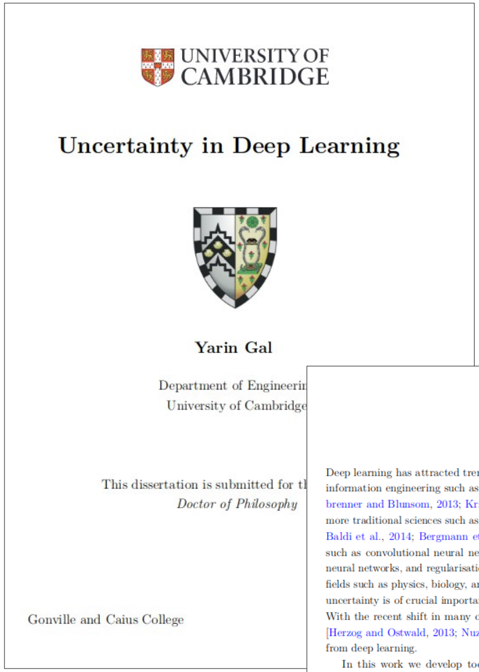
$$P(\mathbf{W}|\mathbf{X},\mathbf{Y}) = \frac{P(\mathbf{X}|\mathbf{Y},\mathbf{W})P(\mathbf{W})}{\int P(\mathbf{X}|\mathbf{Y},\mathbf{W})P(\mathbf{W})d\mathbf{W}}$$

MCMC

Variational Inference

- Alto coste computacional
- Limitaciones en el diseño de arquitecturas

Bayesian deep learning


UNIVERSITY OF CAMBRIDGE

Uncertainty in Deep Learning

Yarin Gal

Department of Engineerin
University of Cambridge

This dissertation is submitted for t
Doctor of Philosophy

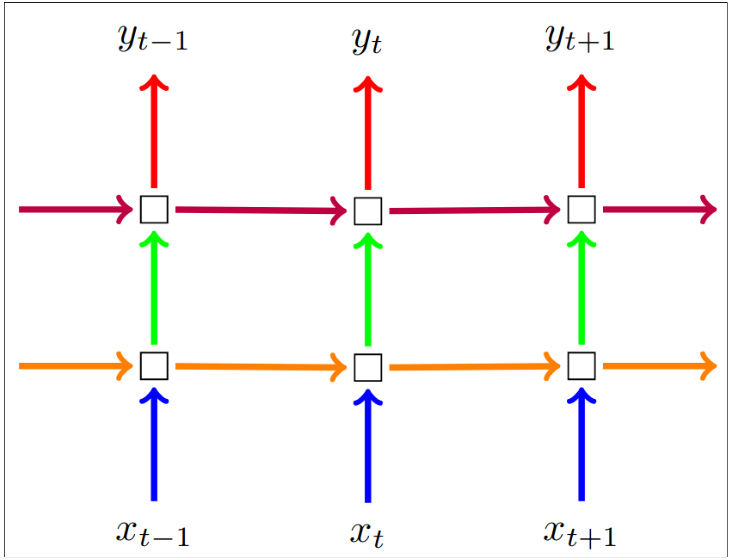
Gonville and Caius College


Abstract

Deep learning has attracted tremendous attention from researchers in various fields of information engineering such as AI, computer vision, and language processing [Kalchbrenner and Blunsom, 2013; Krizhevsky et al., 2012; Mnih et al., 2013], but also from more traditional sciences such as physics, biology, and manufacturing [Anjos et al., 2015; Baldi et al., 2014; Bergmann et al., 2014]. Neural networks, image processing tools such as convolutional neural networks, sequence processing models such as recurrent neural networks, and regularisation tools such as dropout, are used extensively. However, fields such as physics, biology, and manufacturing are ones in which representing model uncertainty is of crucial importance [Ghahramani, 2015; Krzywinski and Altman, 2013]. With the recent shift in many of these fields towards the use of Bayesian uncertainty [Herzog and Ostwald, 2013; Nuzzo, 2014; Trafimow and Marks, 2015], new needs arise from deep learning.

In this work we develop tools to obtain practical uncertainty estimates in deep learning, casting recent deep learning tools as Bayesian models without changing either the models or the optimisation. In the first part of this thesis we develop the theory for such tools, providing applications and illustrative examples. We tie approximate inference in Bayesian models to dropout and other stochastic regularisation techniques, and assess the approximations empirically. We give example applications arising from this connection between modern deep learning and Bayesian modelling such as active learning of image data and data-efficient deep reinforcement learning. We further demonstrate the tools' practicality through a survey of recent applications making use of the suggested techniques in language applications, medical diagnostics, bioinformatics, image processing, and autonomous driving. In the second part of the thesis we explore the insights stemming from the link between Bayesian modelling and deep learning, and its theoretical implications. We discuss what determines model uncertainty properties, analyse the approximate inference analytically in the linear case, and theoretically examine various priors such as spike and slab priors.

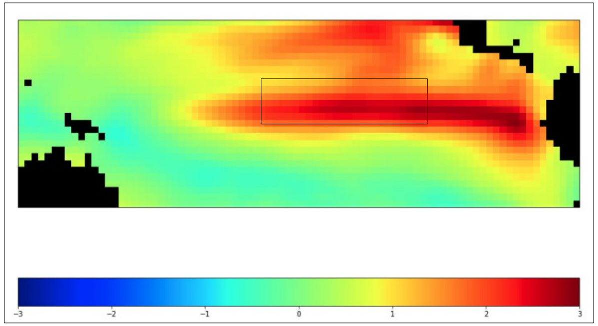
Si se define la función de distribución de aproximación Q del variational inference de una red neuronal bayesiana a partir de una Bernoulli su optimización es equivalente a la realizada por una red neuronal regularizada con dropout

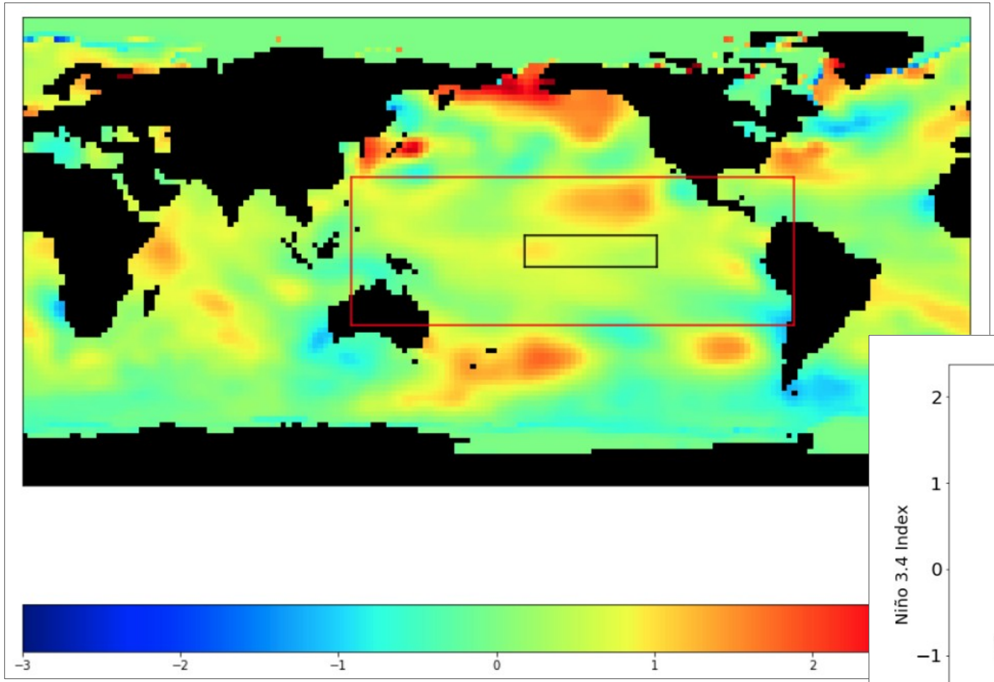


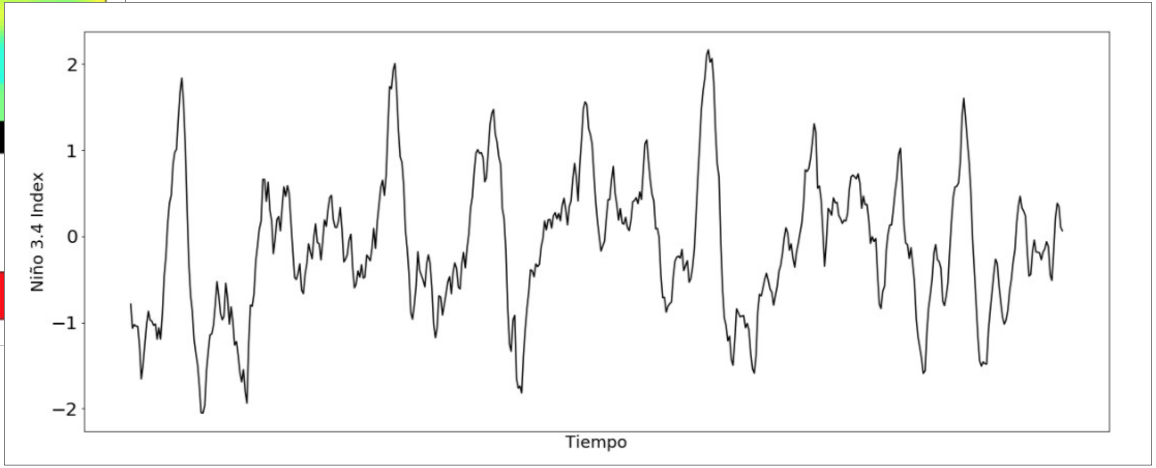
Datos y Metodología

Datos y Metodología

Fuente: NOAA





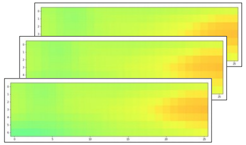


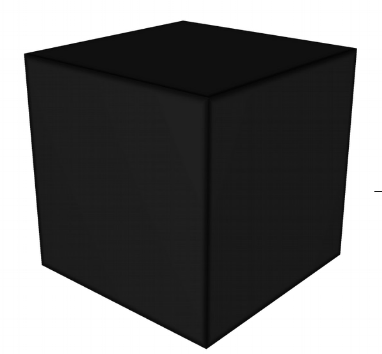
Datos y Metodología

Conjunto de entrenamiento: Enero 1970 – Agosto 2014
Conjunto de test: Febrero 2015 – Diciembre 2016 (forma alterna) → Anomalía ENSO Diciembre 2015

6 meses timeLead

x timeSteps





Índice Niño 3.4

CRPS

$$CRPS(F,h) = \int_R (F(r) - 1\{r \geq h\})^2 dr$$

MSPE/MSE

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

Métodos

Modelos de referencia

Linear DTSM

$$Y_{t,i} = \sum_{j=1}^{n_y} a_{i,j} Y_{t-1,j} + \zeta_{t,i}^{(l)}$$

GQN

$$Y_{t,i} = \sum_{j=1}^{n_y} a_{i,j} Y_{t-1,j} + \sum_{k=1}^{n_y} \sum_{k=1}^{n_y} b_{i,k,l} Y_{t-1,k} Y_{t-1,l} + \zeta_{t,i}^{(q)}$$

E-QESN

Reservoir computing

BAST-RNN

$$Y_t = \boldsymbol{\mu} + \mathbf{V}_1 \mathbf{h}_t + \mathbf{V}_1^2 \mathbf{h}_t^2 + \epsilon_t, \quad \epsilon_t \sim Gau(0, \mathbf{R}_t)$$

$$\mathbf{h}_t = f\left(\frac{\delta}{\mid \lambda_w \mid} \mathbf{W} \mathbf{h}_{t-1} + \mathbf{U} \mathbf{x}_t\right)$$

MCMC

Modelos desarrollados

Dropout p = 0.05, 0.1, 0.2 , 0.5, 0.8
Algoritmo ADAM
Early-stopping sobre un 10% de datos de validación

NN

<u>Versión sin Dropout</u>

Capa de entrada
+
Capa intermedia con 4 neuronas y ReLU
+
Capa Salida (Lineal)

Batch = 64 Learning Rate = 0.001

<u>Versión con Dropout</u>

Dropout en capa de entrada
Learning Rate = 0.0001

LIN

Modelo lineal de Benchmark

Combinación lineal de píxeles

Modelos desarrollados

timeSteps = 2, 12, 24

timeSteps = 12

RNN

<u>Versión sin Dropout</u>

Capa de entrada
+
Capa intermedia con 20 neuronas (ReLU + tanh) y regularizador L2
+
Capa Salida (Lineal)

Batch = 64   Learning Rate = 0.01

<u>Versión con Dropout</u>

Dropout de acuerdo al esquema

timeSteps = 2

LSTM

<u>Versión sin Dropout</u>

Capa de entrada
+
Capa intermedia con 20 neuronas (ReLU + tanh)
+
Capa Salida (Lineal)

Batch = 64   Learning Rate = 0.01

<u>Versión con Dropout</u>

Dropout de acuerdo al esquema

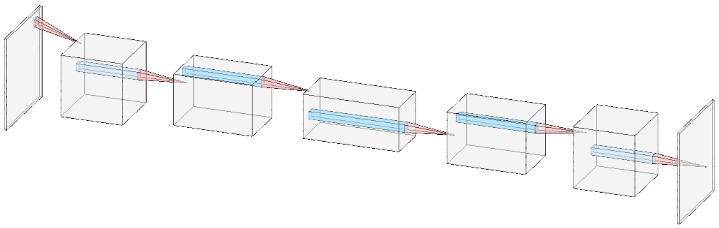
Modelos desarrollados

TimeSteps = 1, 2, 12, 24

timeSteps = 2

Conv AE

<u>Versión sin Dropout</u>



Activación ReLU
Filtros 3x3
12 24 48 24 12

Batch = 64   Learning Rate = 0.01

timeSteps = 2, 12, 24

timeSteps = 12

Conv LSTM

<u>Versión sin Dropout</u>

Capa de entrada
+
Capa intermedia con 12 filtros
+
Capa salida con 1 filtro (Lineal)

Batch = 64   Learning Rate = 0.001

<u>Versión con Dropout</u>

Dropout de acuerdo al esquema

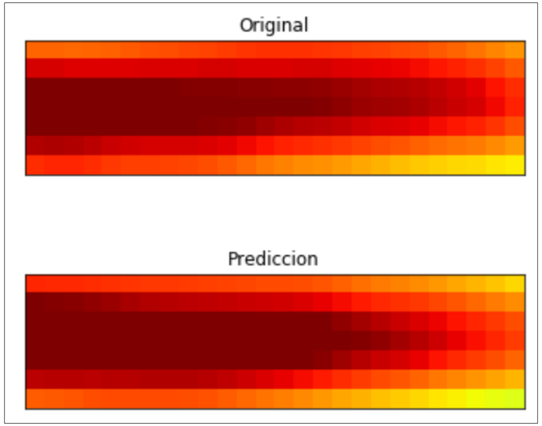
Resultados

Resultados

Tabla 5.1: Comparación en términos de MSPE y CRPS de los distintos modelos ajustados. En negrita los de referencia. La t hace referencia al número de meses utilizados como recurrencia en la capa de entrada. El DP es el valor del hiperparámetro p del dropout.

Modelo	Niño 3.4 MSPE	Niño 3.4 CRPS
BAST-RNN	**0.193**	**0.290**
NN	0.241	-
LSTM t=2	0.259	-
E-QESN	**0.261**	**0.354**
RNN t=12	0.291	-
LIN	0.357	-
RNN t=24	0.4	-
Conv AE t=2	0.401	-
RNN t=2	0.427	-
LSTM t=24	0.438	-
Conv AE t=1	0.44	-
LSTM t=2 w/ DP=5 %	0.491	0.534
RNN t=12 w/ DP=5 %	0.503	0.469
GQN	**0.619**	**0.538**
Conv AE t=12	0.757	-
NN w/ DP=5 %	0.766	0.689
Lin. DSTM	**0.785**	**0.699**
LSTM t=12	0.839	-
Conv LSTM t=12	0.874	-
Conv AE t=24	0.936	-
Conv LSTM t=2	1.015	-
Conv LSTM t=12 w/ DP=5 %	1.408	0.94
Conv LSTM t=24	1.801	-

Resultados



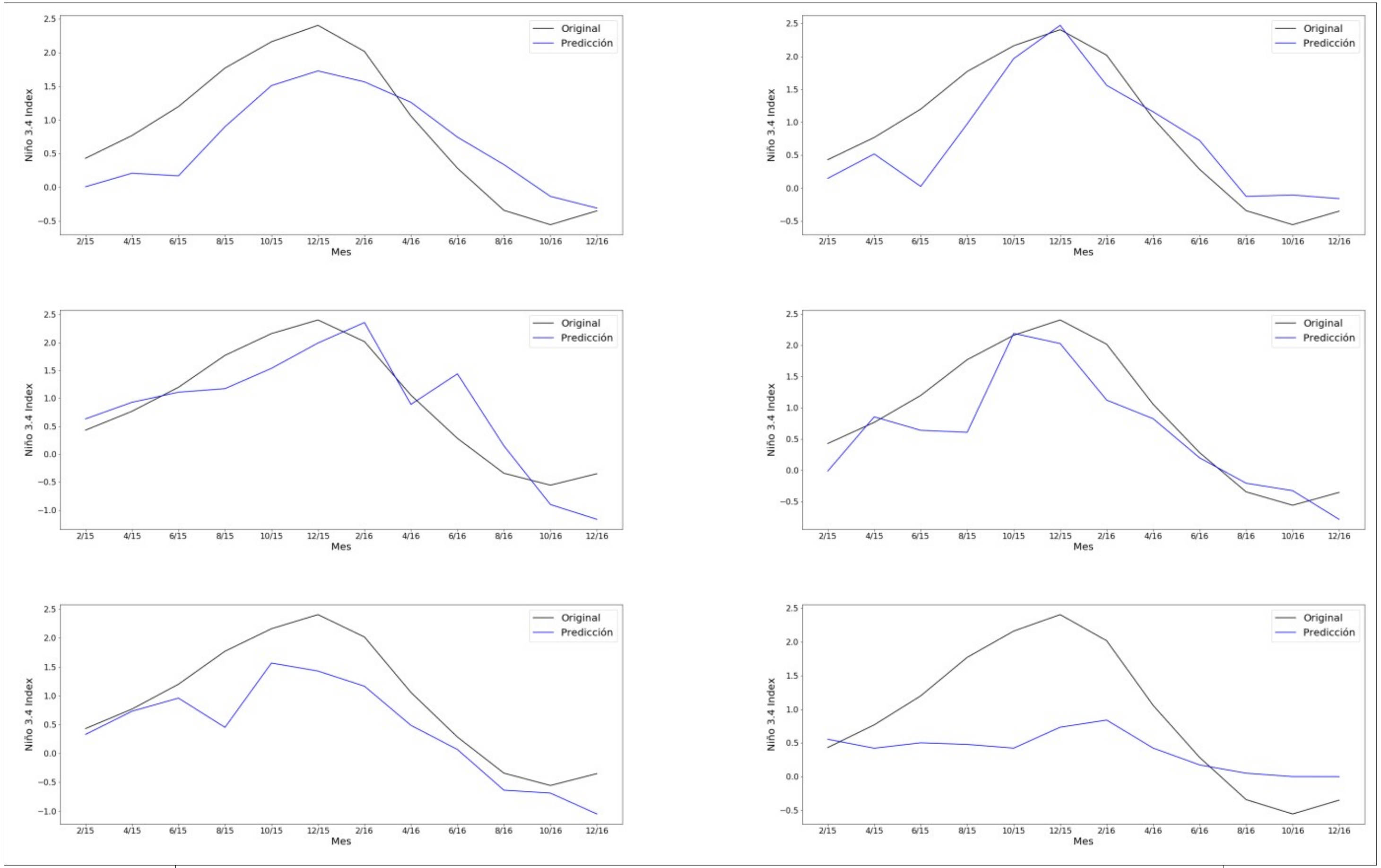


Figura 5.1: Predicciones respecto a los valores originales. En la columna de la izquierda de arriba a abajo se muestran las correspondientes a la red lineal, red recurrente y autoencoder convolucional. En la columna de la derecha de arriba a abajo se muestran las de la red densa, LSTM y LSTM convolucional.

Resultados

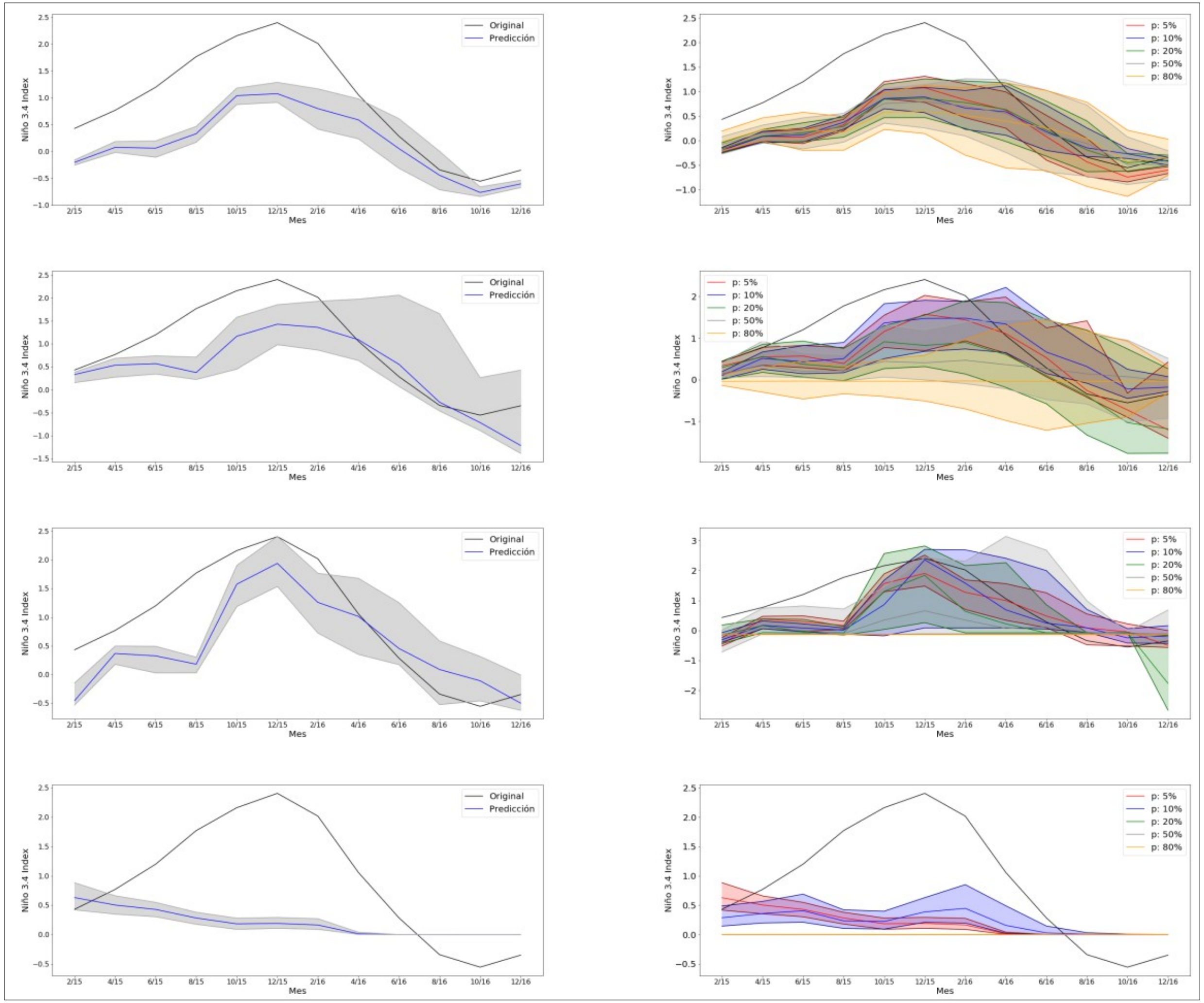


Figura 5.3: En la columna de la izquierda de arriba a abajo se muestran los intervalos de confianza con $p = 5\,\%$ para la red densa, red recurrente, LSTM y LSTM convolucional. En la columna de la derecha los intervalos de confianza para distintos valores de p

Resultados

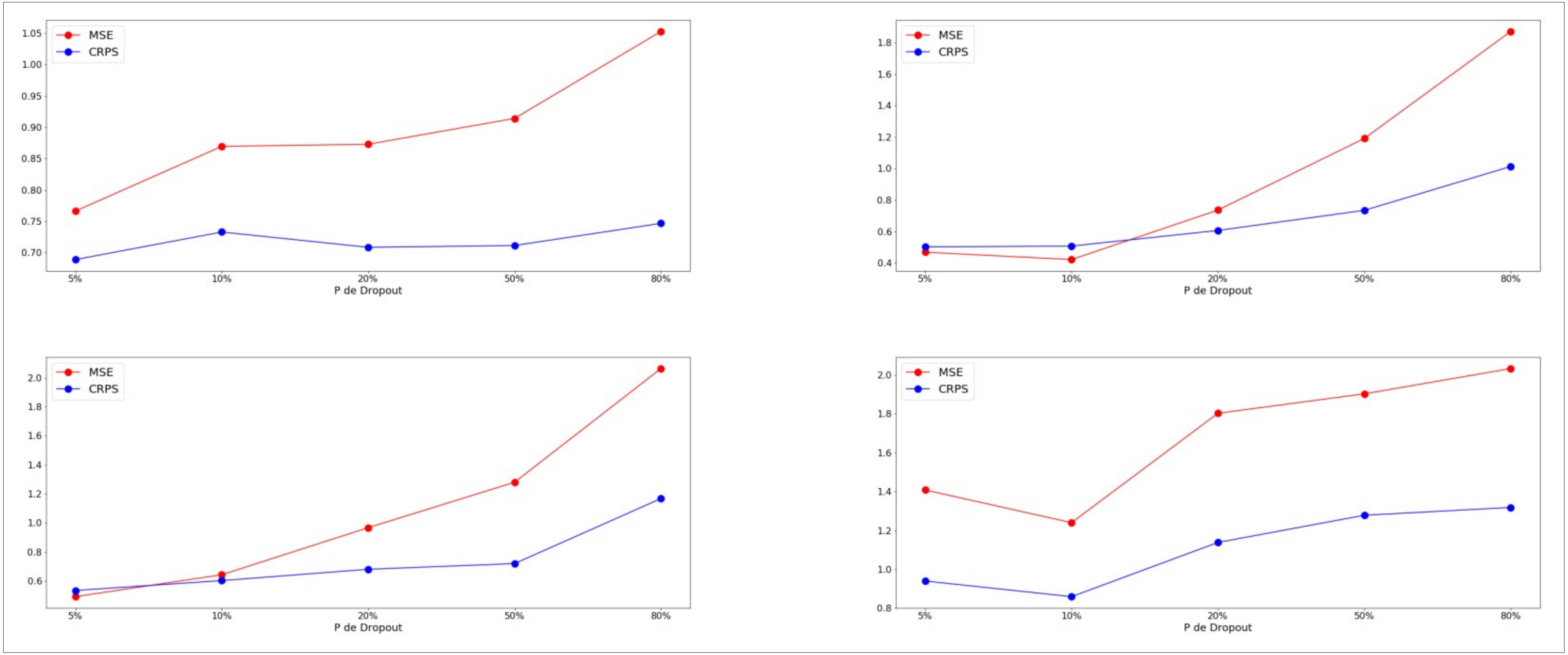


Figura 5.4: Evolución del MSE Y CRPS en función de la p del dropout. En la columna izquierda de arriba a abajo la correspondiente a la red densa y LSTM. En la columna derecha la de la red recurrente y LSTM convolucional

Conclusiones

Conclusiones

- Los modelos desarrollados superan al Benchmark lineal
- Los modelos sencillos son los que mejores resultados obtienen
- La red neuronal densa consigue por primera vez predecir correctamente la anomalía de Diciembre de 2015
- Los intervalos de confianza están muy condicionados al valor de *p*
- Los mejores intervalos son los de los modelos recurrentes

Trade-off entre la capacidad de aprendizaje y la cuantificación de la incertidumbre

¿Preguntas?