|  |  |
| --- | --- |
|  | DSNBoost |
|  |  |
| V.4 | Report 10/12/2015 |
|  | Author:  MANUEL MONTOYA CATALÁ |

[1. Introduction 2](#_Toc437508223)

[1.1 Algoritmo de Aprendizaje de cada capa 3](#_Toc437508224)

[1.2 Parámetros de los experimentos 3](#_Toc437508225)

[2. Abalone 4](#_Toc437508226)

[1.3 Efecto de Nepoch 5](#_Toc437508227)

[3. Kwok 6](#_Toc437508229)

[3.1 Efecto de Nepoch 7](#_Toc437508231)

[4. Image 8](#_Toc437508232)

[3.2 Efecto de Nepoch 9](#_Toc437508233)

[5. Waveform 10](#_Toc437508235)

[4.1 Efecto de Nepoch 11](#_Toc437508236)

[6. Conclusiones DSNboost 12](#_Toc437508238)

[5.1 Líneas futuras 13](#_Toc437508239)

# Introduction

En este report se analizarán cuantitativa y cualitativamente las propiedades del algoritmo DSNBoost. Se describirá su comportamiento respecto a 5 parámetros:

* Número de Capas. nL
* Número de Neuronas Ocultas. Nh
* Número de Épocas Nepoch
* Valor de alpha
* Valor de beta

También se describirá en menor medida su comportamiento en función de los parámetros de entrenamiento de las capas:

* Constante inicialización.
* Ratio entre constante aprendizaje de capa oculta y de salida.

Se han utilizado 3 formas diferentes de aprendizaje de las capas:

* Report 1:
* Report 2:
* Report 3:

Este report es un **review de los 3 reports** anteriores con el objetivo de sacar conclusiones sobre cómo afectan los parámetros a la accuracy y cómo se afectan entre sí para poder inferir las mejores líneas futuras a seguir.

Primeramente se analizarán los resultados obtenidos en los 3 reports anteriores para cada base de datos por separado, describiendo el comportamiento del algoritmo para cada una de ellas en función de los parámetros utilizados, viendo principalmente la evolución de la accuracy del error de test con nL, nH y alpha.

## Algoritmo de Aprendizaje de cada capa

El algoritmo de aprendizaje **para cada capa** es backpropagation. Los parámetros utilizados son:

* Inicialización de pesos: Variable aleatoria uniforme entre -ro y ro donde:

Se han realizado experimentos para

* Constante de aprendizaje Inicial: La constante de aprendizaje inicial es diferente para la capa oculta y para la capa de salida. La capa oculta tiene una constante inicial de mientras que la de la salida tiene una constante S veces menor .

Se han realizado experimentos para

* Criterio de parada: Se ha utilizado un criterio de parada que no requiera utilizar error de validación (como se pidió). El algoritmo tiene un **número de épocas fijo** y se han realizado experimentos para

La constante de aprendizaje **decae linealmente** **a 0** durante el número de épocas.

* La función de coste a minimizar es el error cuadrático medio, MSE.
* La función de activación de las **neuronas ocultas** es tanh.
* La función de activación de la **neurona de salida** es tanh.

**Dado que el entrenamiento es el mismo independientemente de nH vamos a hacer la suposición de que:**

* **Mayor capacidad expresiva 🡪 Overfitting.**
* **A mayor nH: Menos se entrenan las neuronas 🡪 Underfitting**

**Estas dos componentes lucharán y darán lugar a un valor de nH intermedio.**

Podríamos también suponer que, por lo menos para AODSN:

* El sistema mejorará con nL mientras mejore O y no se produzca overfitting.
* Si el resultado de O empieza a decaer debido al underfitting, se acabó lo que se daba.

## Parámetros de los experimentos

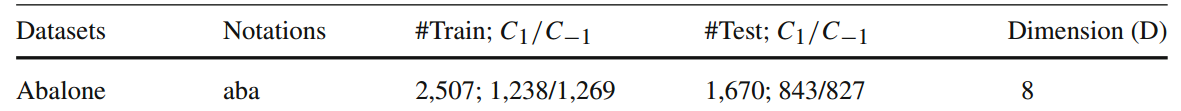
Para el cálculo del **error omnisciente** se han llevado a cabo **20 realizaciones** para cada conjunto de parámetros.

En las gráficas se visualiza la **accuracy** del algoritmo, es decir, el porcentaje de acierto.

Siempre que hablemos de error de entrenamiento o de test nos estamos refiriendo a los errores habiendo entrenado con todo el train y después mirando el error de test.

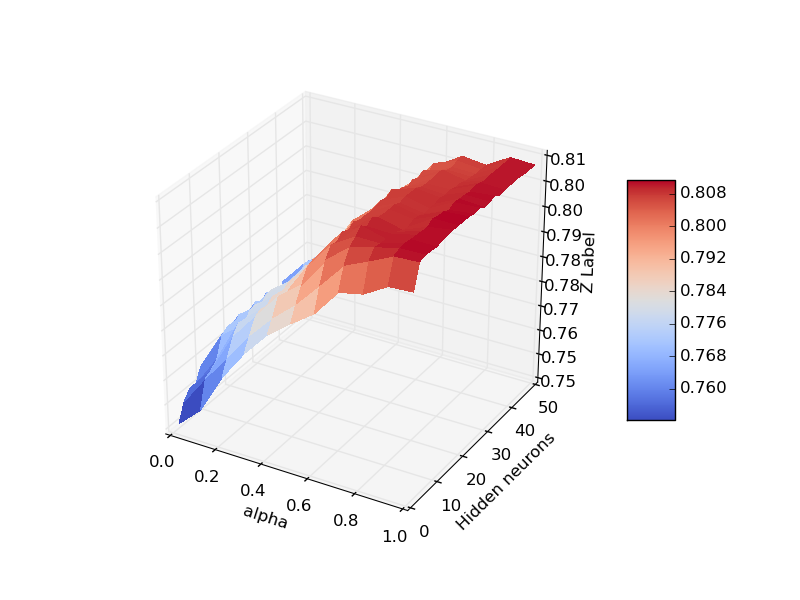
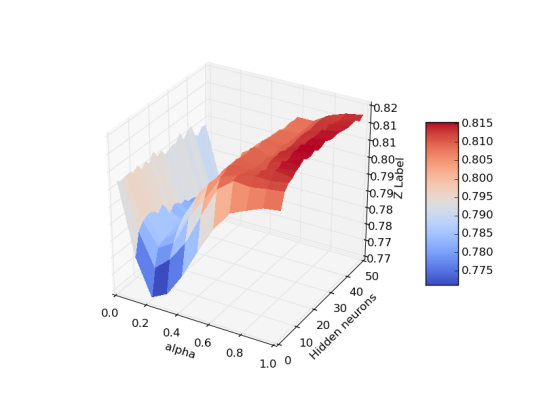
# Abalone

A continuación se describirán las propiedades de la BBDD Abalone y los resultados obtenidos para los reports R2 y R3.

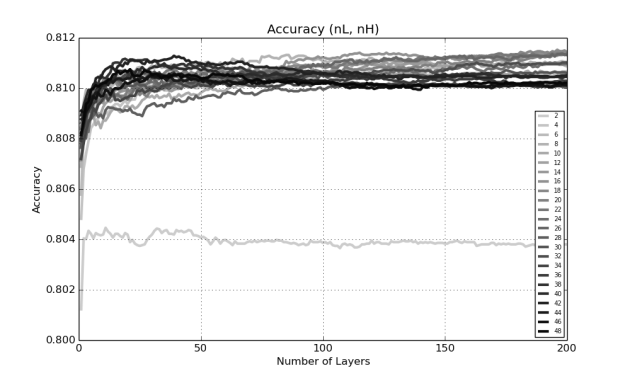
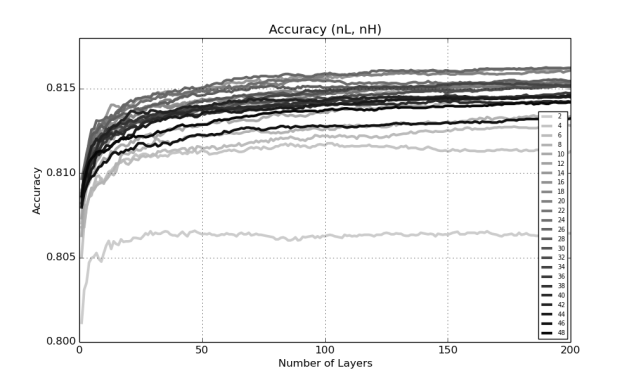


**El valor inicial de gama = 0.482, AODSN evolución descendente.**

Las siguientes gráficas muestran la media de accuracy de tst para R2 y R3 respectivamente utilizando 100 épocas.



Las siguientes 2 gráficas muestran el error de test para , y R2, R3 respectivamente.



De acuerdo a estas gráficas Abalone es una base de datos que requiere:

* Un **valor de alpha alto**, es decir, no requiere énfasis. El valor de alpha suele suele estar en 0.9 y 1.
* Número de neuronas medio 10-20.

El comportamiento de la accuracy respecto a alpha es homólogo para . parece remontar la accuracy en pero nada más.

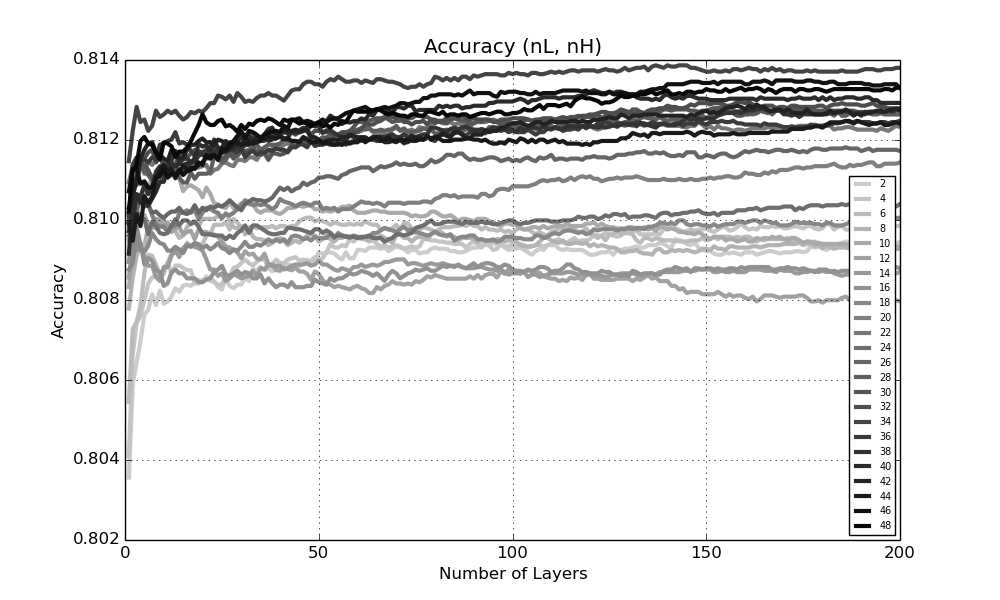
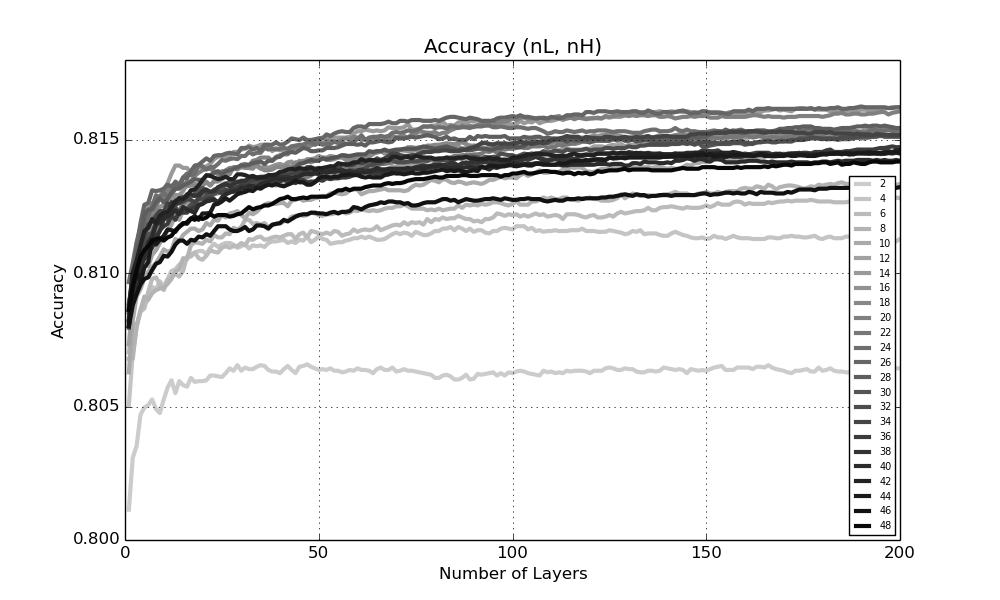
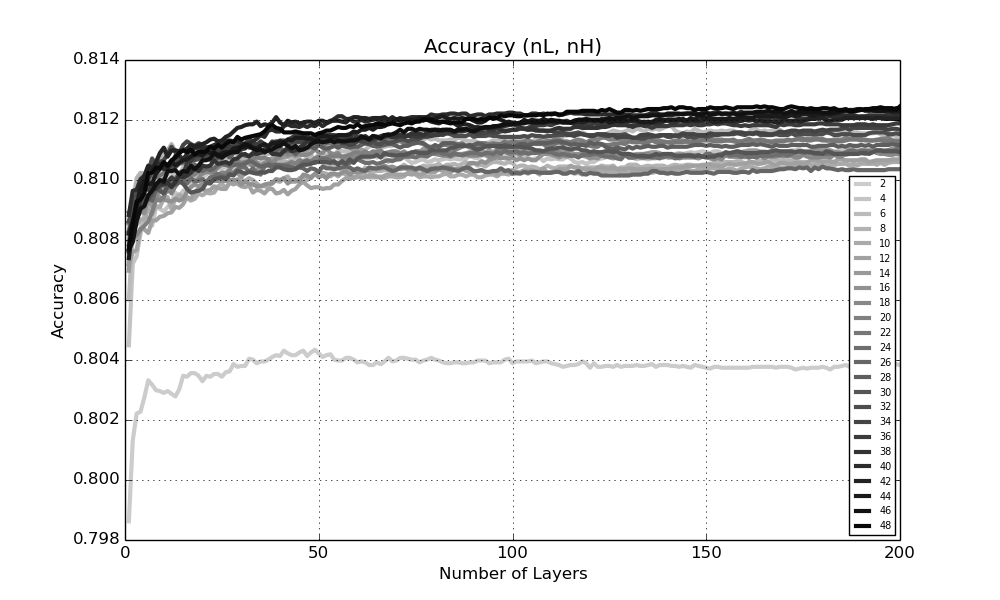
La accuracy se desploma para valores bajos de alpha. Tanto para beta = 1 como beta = 0.

La evolución de R2 con nL es buena mientras que la de R3 se estanca para valores pequeños de nH y nL. Deberían buscarse otros parámetros de entrenamiento que hagan mejor la evolución.

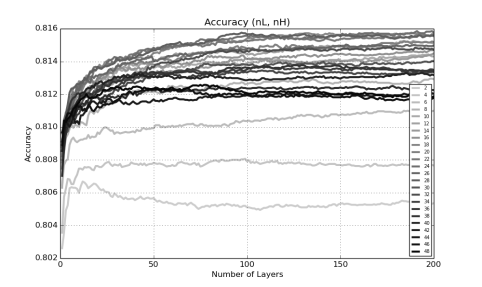
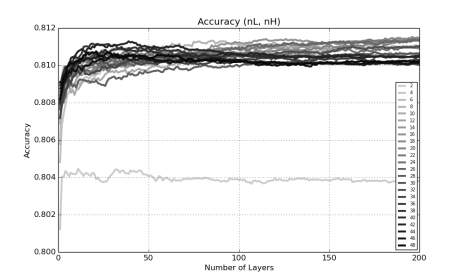
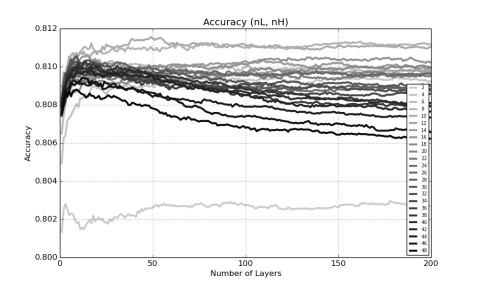
## Efecto de Nepoch

A continuación mostraremos cómo evoluciona la accuracy de test en función de nL para diferentes Nepoch con el objetivo de ver cómo afecta este parámetro al algoritmo.

Las siguientes 3 gráficas muestran la accuracy de test para el entrenamiento R2, y respectivamente



Las siguientes 3 gráficas muestran la accuracy de test para el entrenamiento R3, y respectivamente.

****

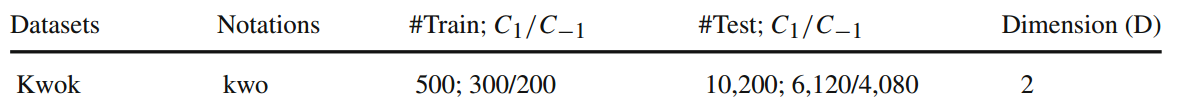
El entrenamiento afecta decisivamente a la evolución y valor final de la accuracy.

* **A mejor entrenamiento, más diferencia hay en la accuracy con nH**. Si el entrenamiento es malo, nH no importa (podría ser un indicador). Hay que desconfiar cuando el resultado es muy similar para diferente número de neuronas.
* Para R2 parece favorecerse un mayor número de neuronas independientemente de Nepoch mientras que para R3 parece favorecerse un número menor.
* Los resultados parecen indicar que R3 tiene **underfitting**. Probablemente debido a una saturación prematura del entrenamiento (comparado con R2).



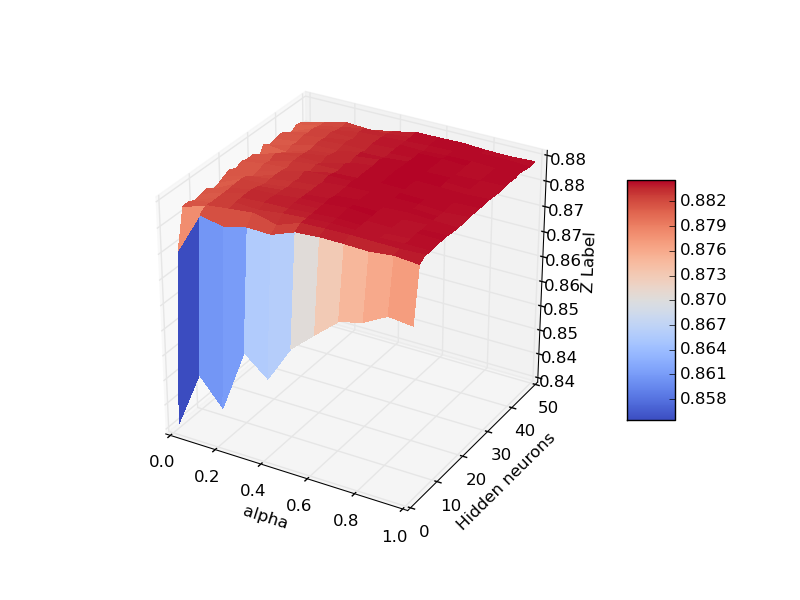
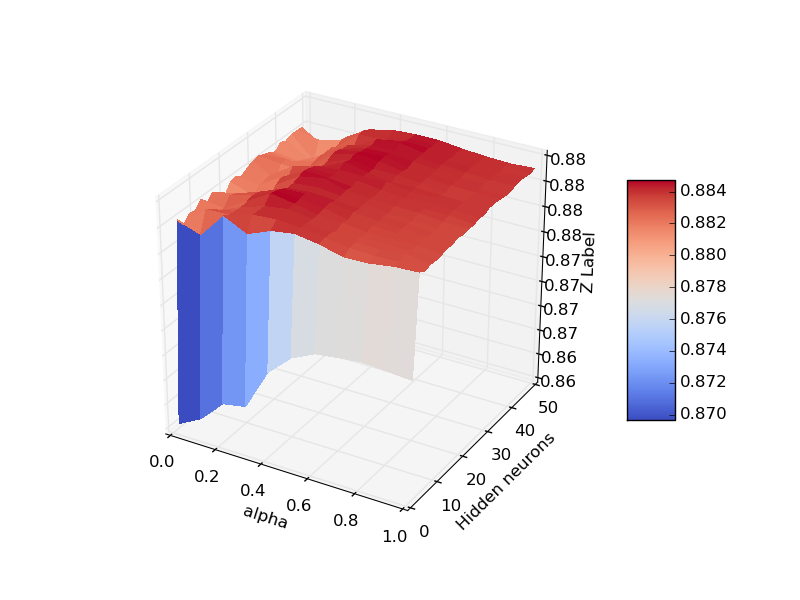
# Kwok

A continuación se describirán las propiedades de la BBDD Kwok y los resultados obtenidos para los reports R2 y R3.

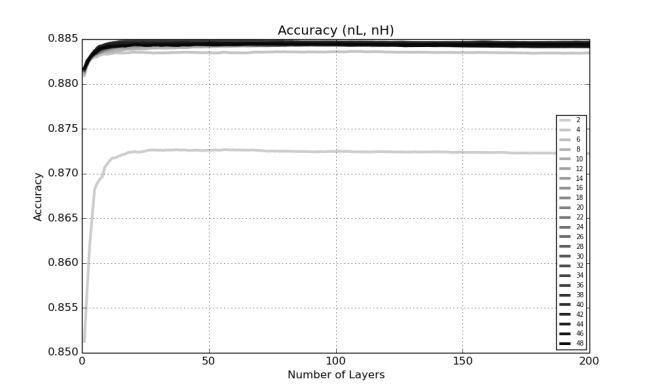
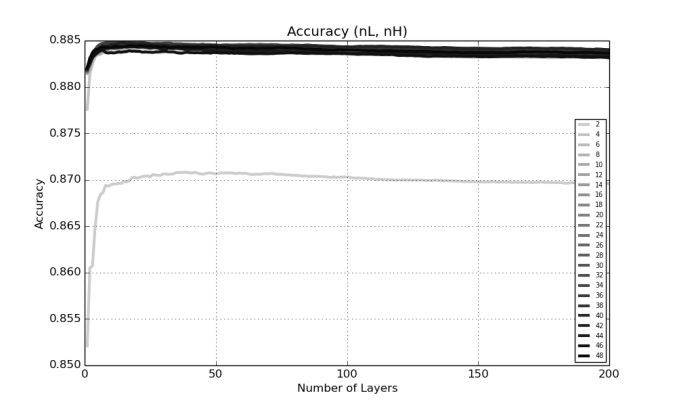


**El valor inicial de gama = 0.9, AODSN evolución descendente.**

Las siguientes gráficas muestran la media de accuracy de tst para R2 y R3 respectivamente utilizando 100 épocas.



Las siguientes 2 gráficas muestran el error de test para , y R2, R3 respectivamente.



De acuerdo a estas gráficas Kwok es una base de datos que requiere:

* Un **valor de alpha medio**. El valor de alpha suele estar en 0.4 y 0.6. Además se mantiene constante para valores de nH y alpha crecientes.
* Número de neuronas bastante indiferente pero alto 30 – 46.

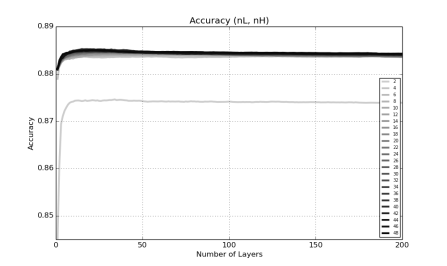
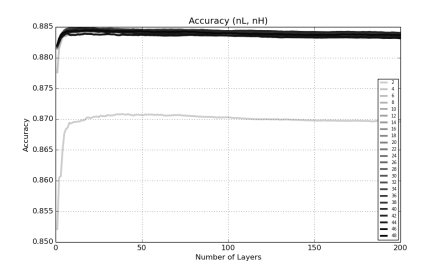
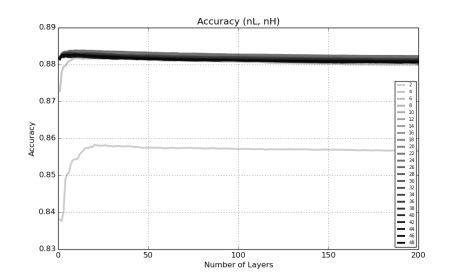
La accuracy no varía significativamente conforme aumentamos nH y alpha. Tanto para beta = 1 como beta = 0. Ambas evoluciones de R2 y R3 se estancan muy tempranas con nL y nH.



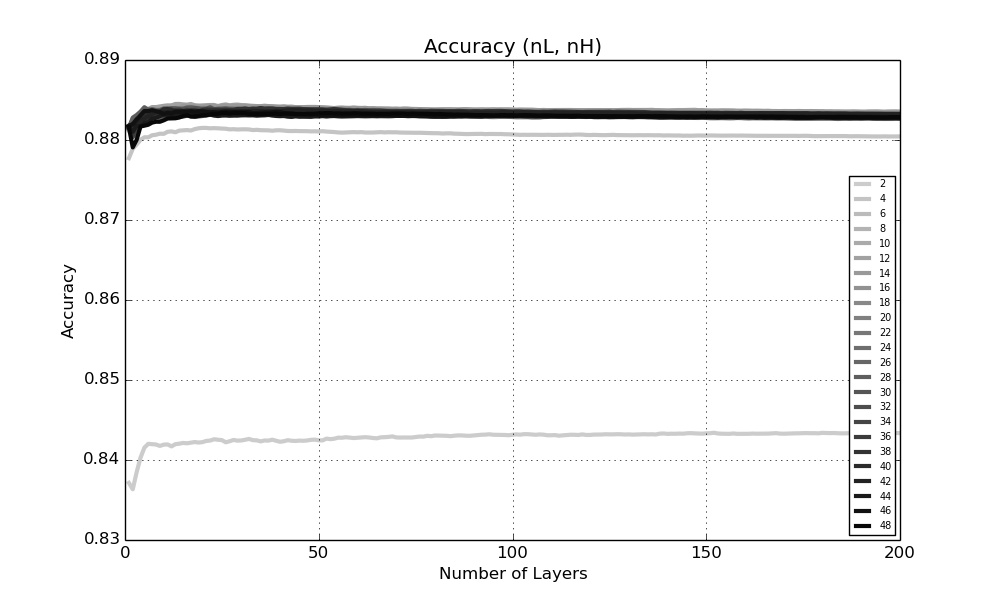
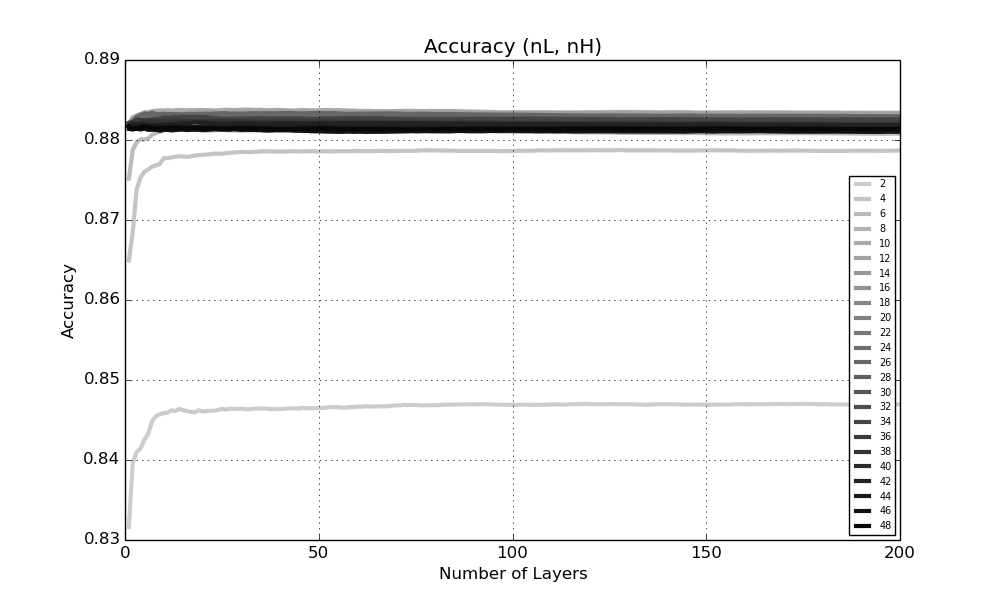
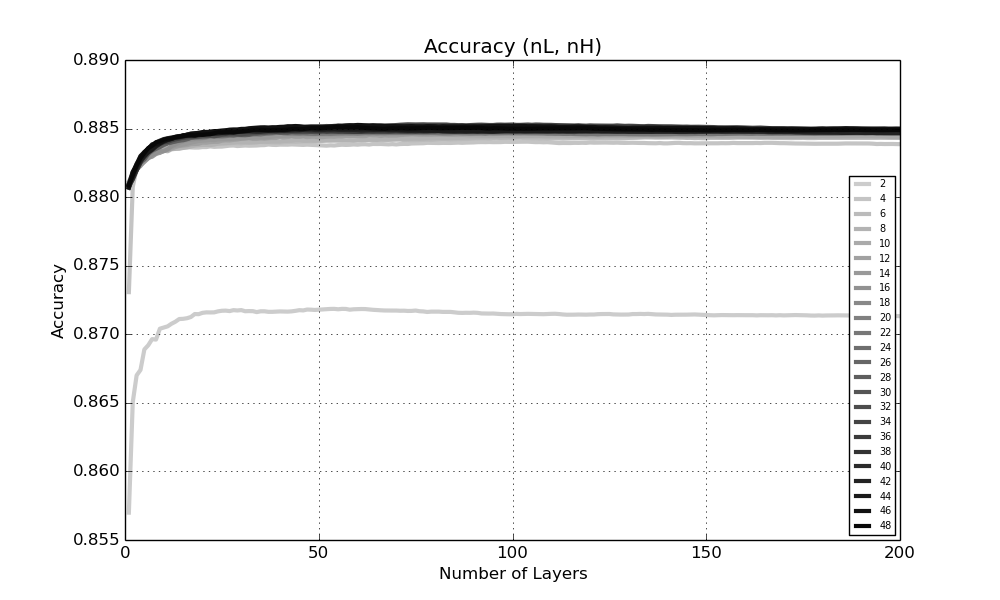
## Efecto de Nepoch

A continuación mostraremos cómo evoluciona la accuracy de test en función de nL para diferentes Necpoch con el objetivo de ver cómo afecta este parámetro al entrenamiento.

Las siguientes 3 gráficas muestran la accuracy de test para el entrenamiento R2, y respectivamente

****

Las siguientes 3 gráficas muestran la accuracy de test para el entrenamiento R3, y respectivamente.

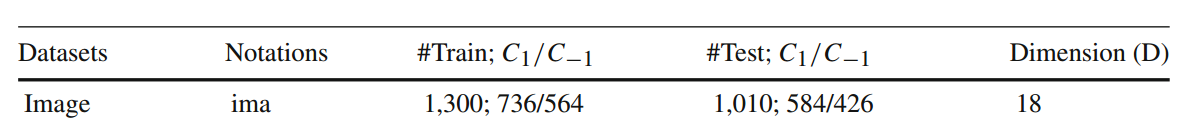


El entrenamiento no afecta decisivamente a la evolución y valor final de la accuracy.

* Apenas evoluciona el error con nL, tan solo en las primeras 20 nL como mucho.
* Podríamos desconfiar y deducir que el entrenamiento no es el adecuado y por eso el error no varía mucho con nH.
* La accuracy suele bajar lentamente con nL después del incremento inicial. Underfitting.

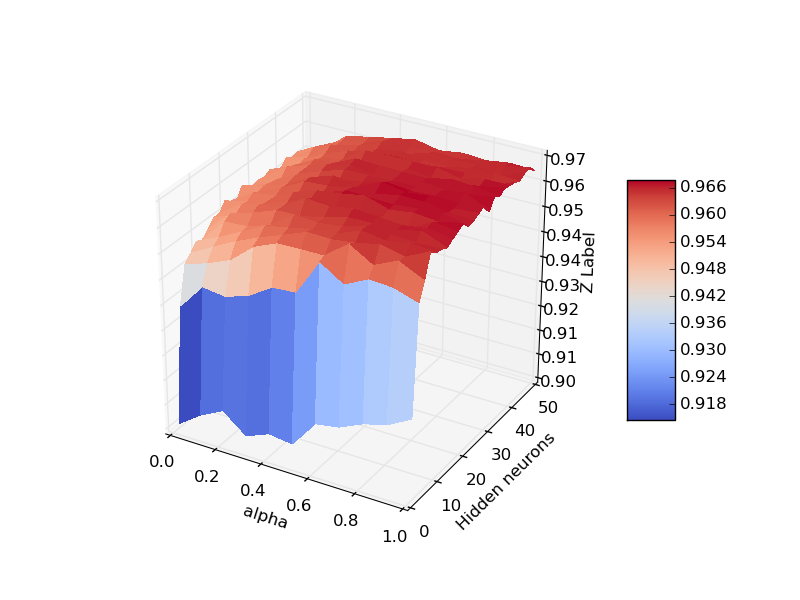
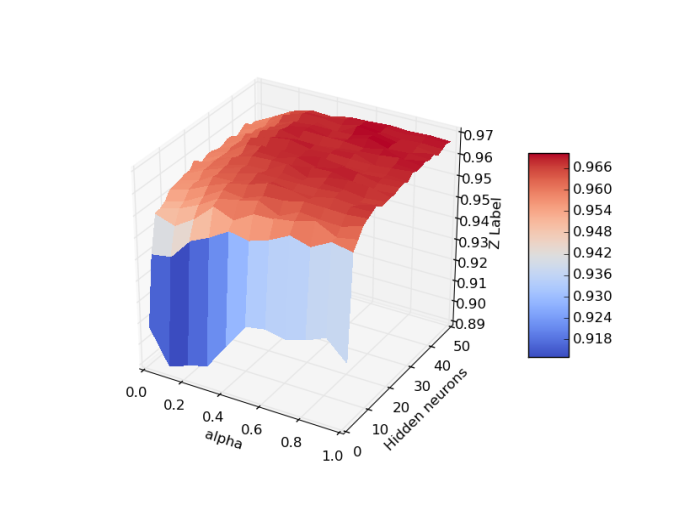
# Image

A continuación se describirán las propiedades de la BBDD Image y los resultados obtenidos para los reports R2 y R3.

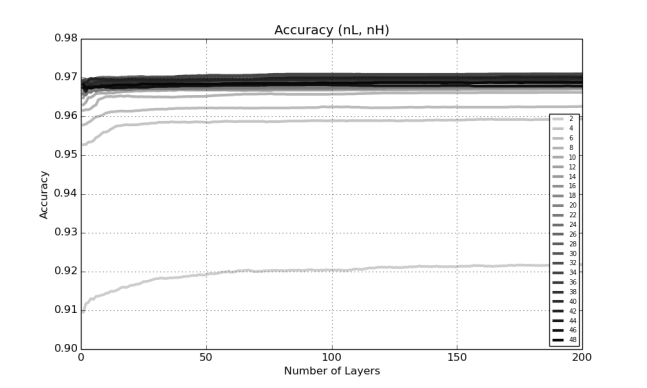
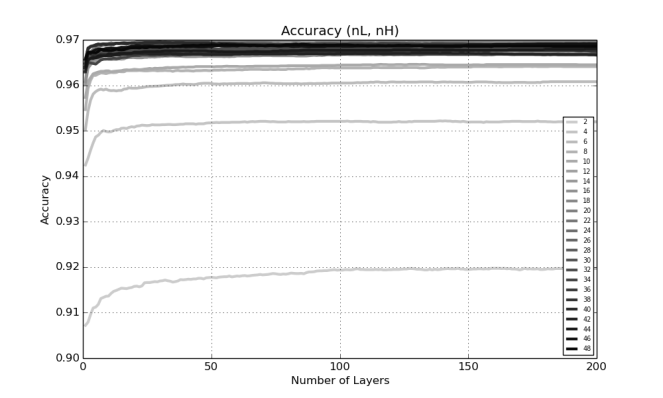


**El valor inicial de gama = 1.5, AODSN evolución ascendente**

Las siguientes gráficas muestran la media de accuracy de tst para R2 y R3 respectivamente utilizando 100 épocas.



Las siguientes 2 gráficas muestran el error de test para , y R2, R3 respectivamente.



De acuerdo a estas gráficas Image es una base de datos que requiere:

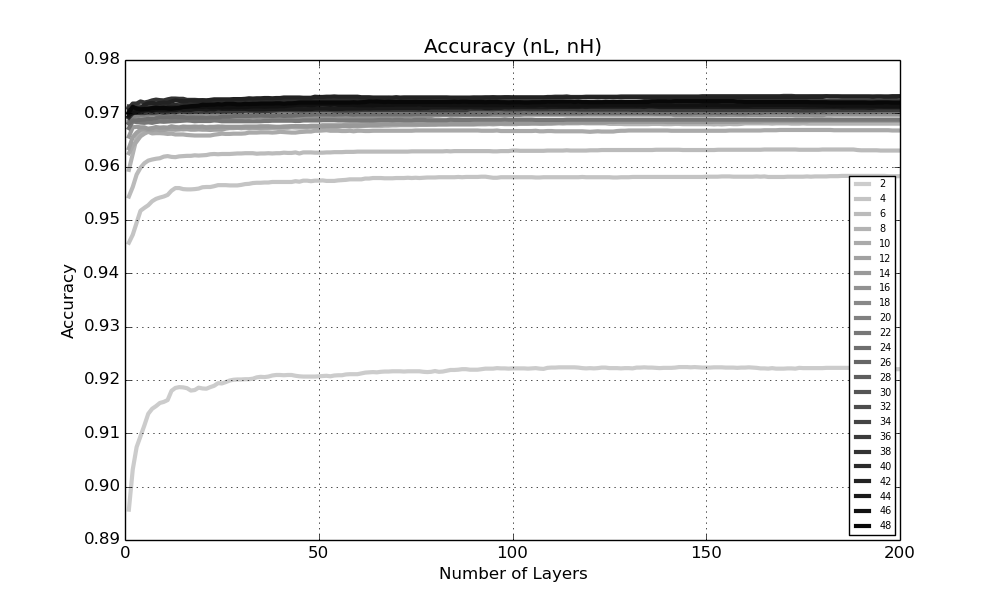
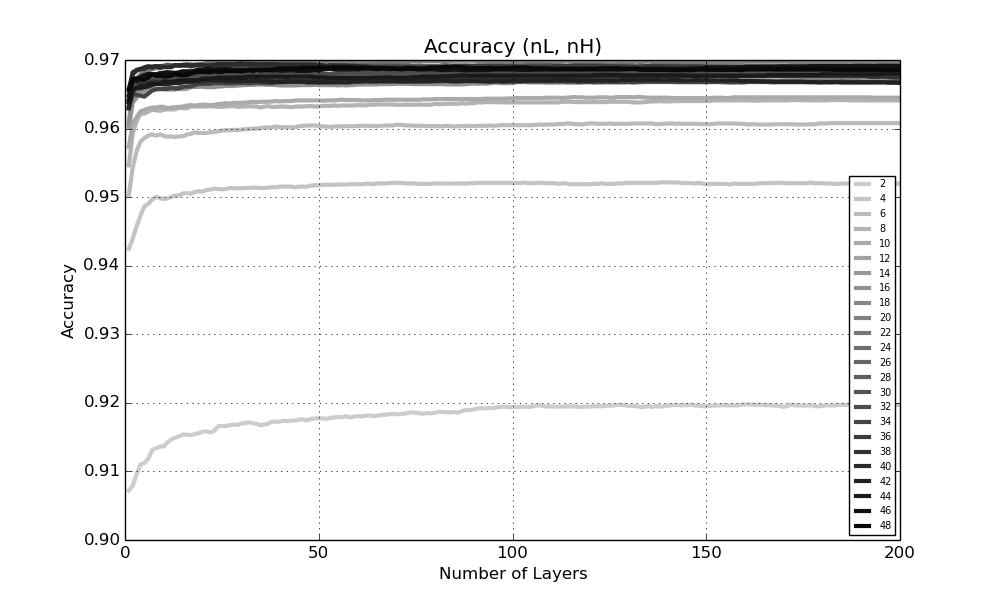
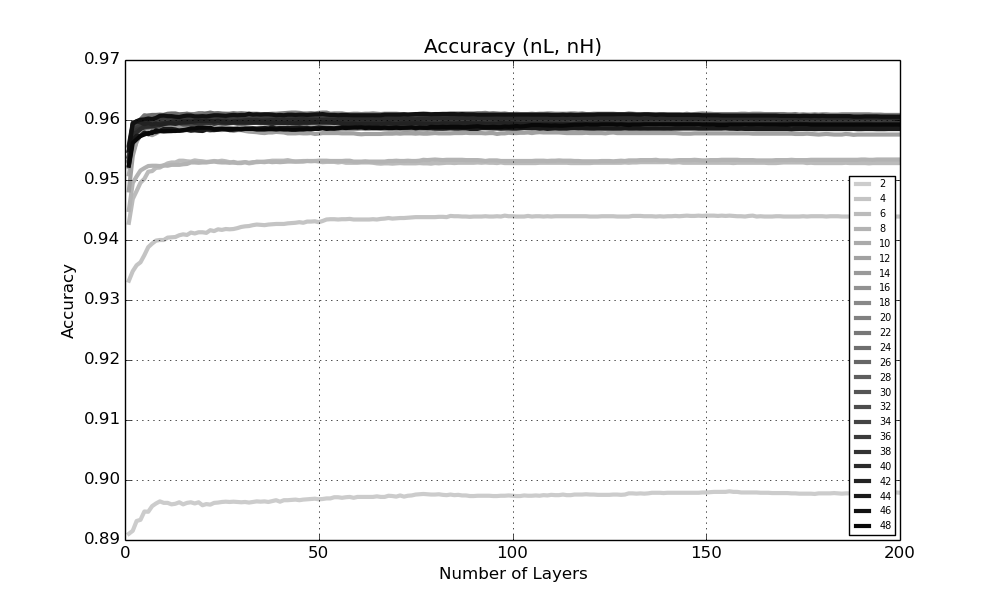
* Un **valor de alpha medio**. El valor de alpha suele estar en 0.5. Además tiende a valores de alpha y nH más grandes.
* Número de neuronas bastante indiferente pero alto 30 – 46.

La accuracy no varía significativamente conforme aumentamos nH y alpha. Tanto para beta = 1 como beta = 0. Ambas evoluciones de R2 y R3 se estancan muy tempranas con nL y nH.

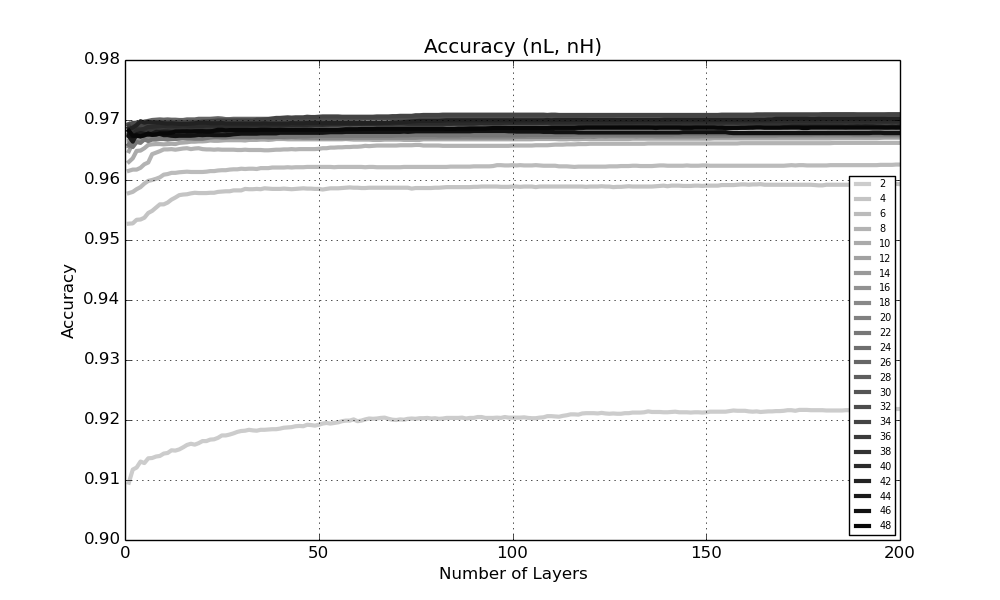
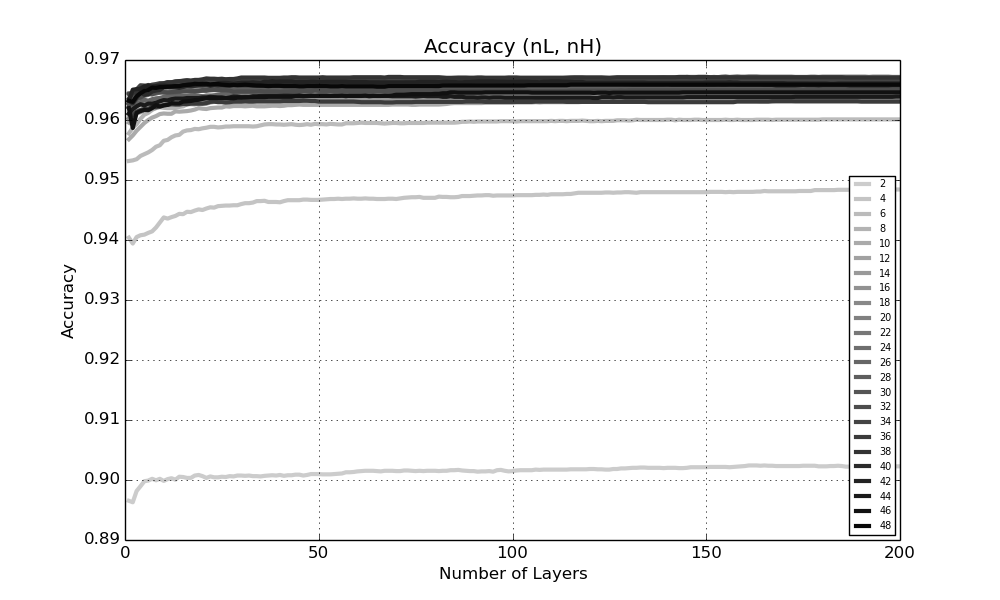
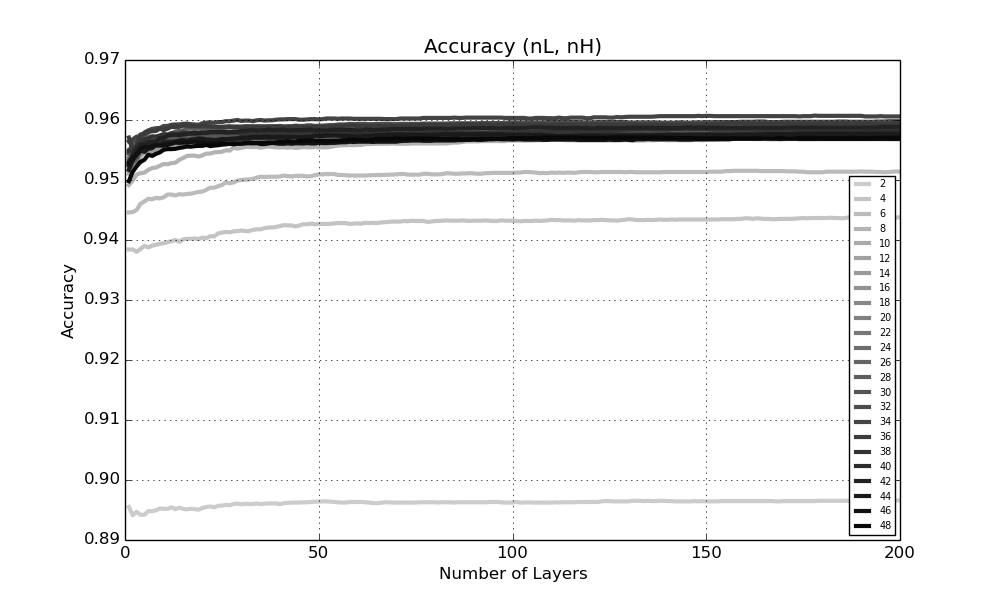
## Efecto de Nepoch

A continuación mostraremos cómo evoluciona la accuracy de test en función de nL para diferentes Necpoch con el objetivo de ver cómo afecta este parámetro al entrenamiento.

Las siguientes 3 gráficas muestran la accuracy de test para el entrenamiento R2, y respectivamente.



Las siguientes 3 gráficas muestran la accuracy de test para el entrenamiento R3, y respectivamente.



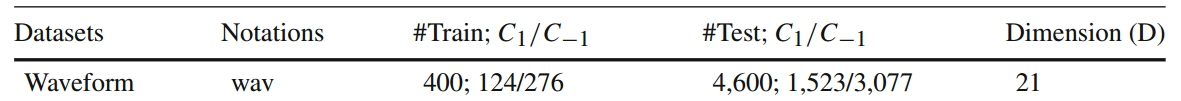
El entrenamiento no afecta decisivamente a la evolución y valor final de la accuracy.

* Apenas evoluciona el error con nL, tan solo en las primeras 20 nL como mucho.
* Podríamos desconfiar y deducir que el entrenamiento no es el adecuado y por eso el error no varía mucho con nH.
* Cuantas más épocas, mejores resultados. Aumentar el número de épocas.



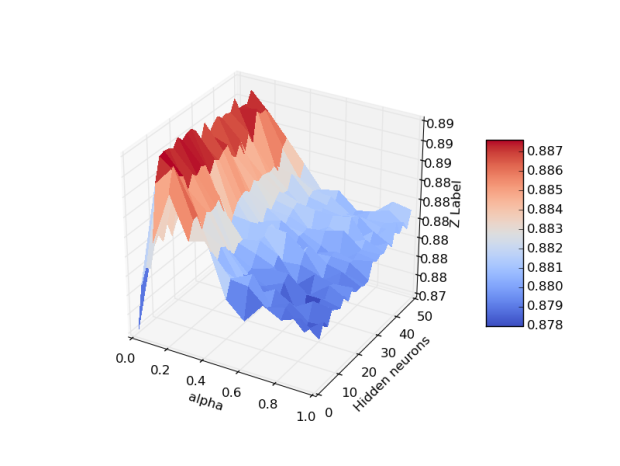
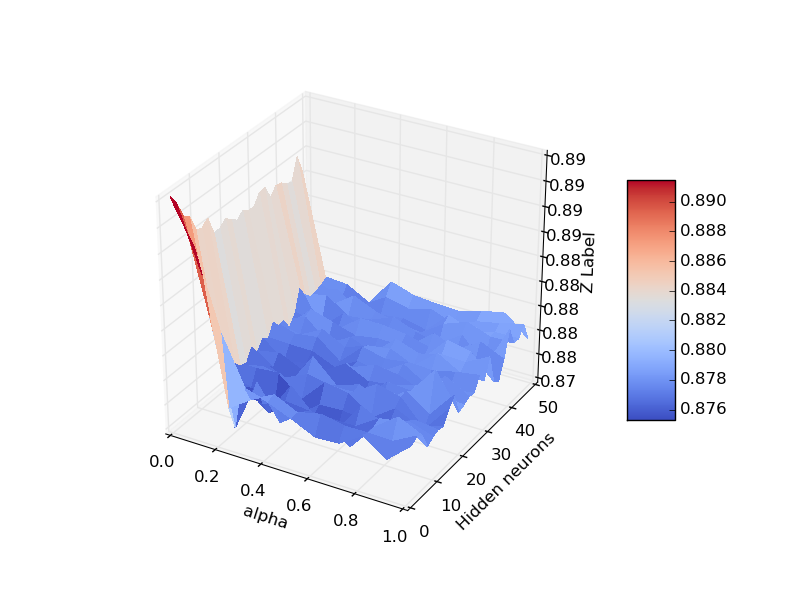
# Waveform

A continuación se describirán las propiedades de la BBDD Abalone y los resultados obtenidos para los reports R2 y R3.

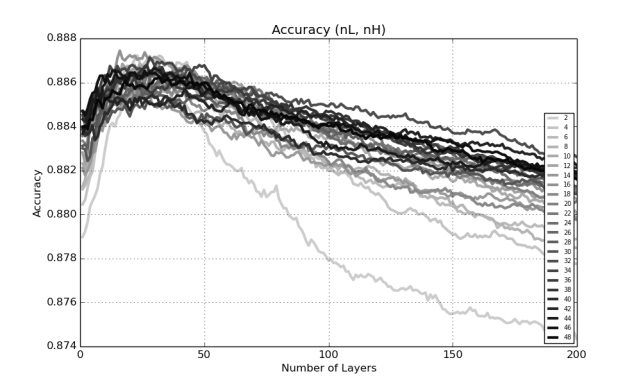
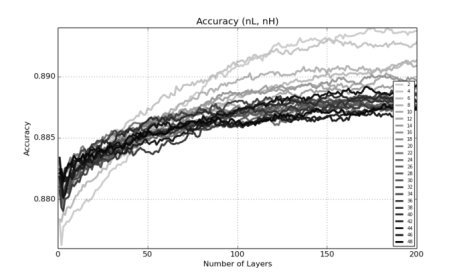


**El valor inicial de gama = 1.4, AODSN evolución ascendente.**

Las siguientes gráficas muestran la media de accuracy de tst para R2 y R3 respectivamente utilizando 100 épocas.



Las siguientes 2 gráficas muestran el error de test para , y R2, R3 respectivamente.



De acuerdo a estas gráficas Waveform es una base de datos que requiere:

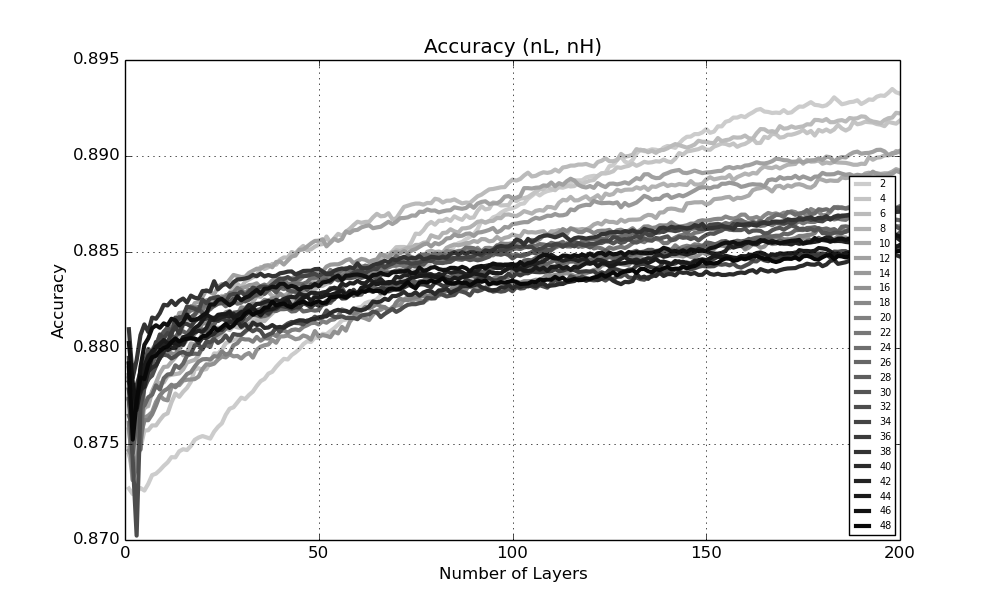
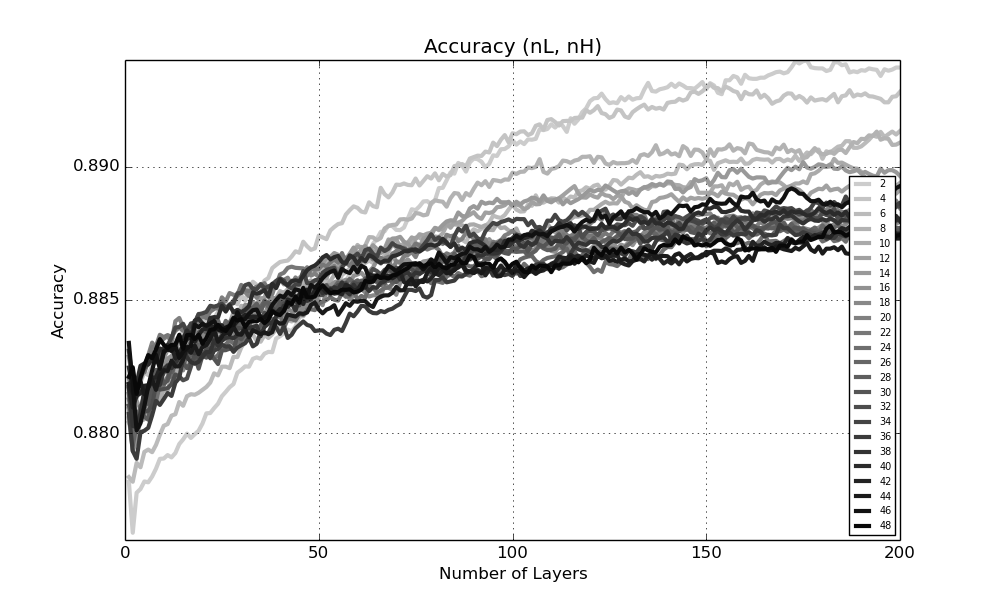
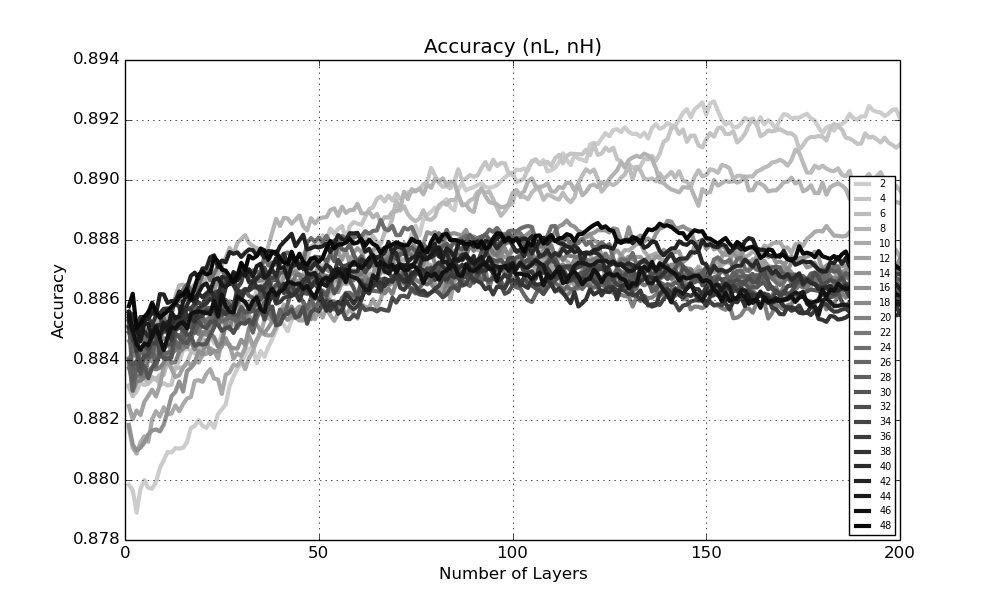
* Un **valor de alpha pequeño**. El valor de alpha suele estar en 0-0.1.
* Número de neuronas bastante desde valores bajos 5 – 46.

La accuracy no varía significativamente conforme aumentamos nH, nL y alpha.

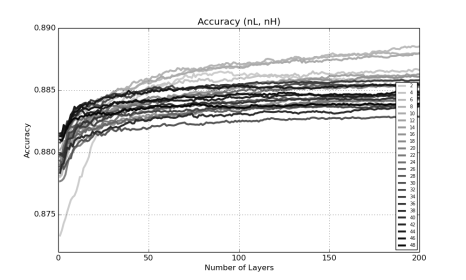
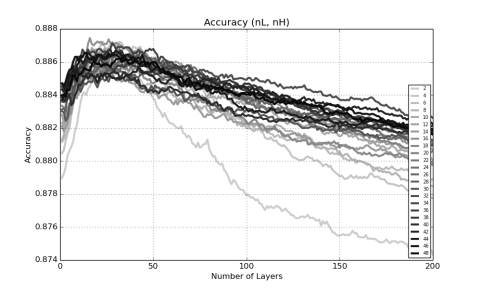
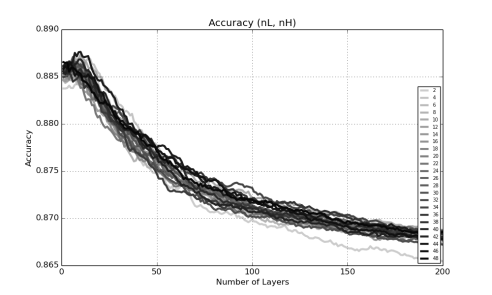
## Efecto de Nepoch

A continuación mostraremos cómo evoluciona la accuracy de test en función de nL para diferentes Necpoch con el objetivo de ver cómo afecta este parámetro al entrenamiento.

Las siguientes 3 gráficas muestran la accuracy de test para el entrenamiento R, y respectivamente.



Las siguientes 3 gráficas muestran la accuracy de test para el entrenamiento R3, y respectivamente.



El entrenamiento afecta decisivamente a la evolución y valor final de la accuracy.

* Para R2, conforme aumentamos Nepoch, la evolución de la accuracy con nL deja tener forma de parábola y se linealiza. Los resultados son buenos independientemente de Nepoch pero según parece, para Nepoch = 200, si aumentamos el número de nL, aumentaría la accuracy. **Conforme aumenta Nepoch, la accuracy final es peor (overfitting) pero luego mejora más con nL.**
* Para R3 conforme aumenta Nepoch, la evolución de la accuracy de tst mejora, pasando de ser descendente a ascendente. Por lo que partimos de un overfitting inicial mayor pero con el underfitting del apilamiento, el sistema generaliza bien. Cuando Nepoch es bajo, se tiene demasiado underfitting.



# Conclusiones DSNboost

Vamos a intentar sacar conclusiones del algoritmo y posibles variaciones del mismo para mejorar en base a estas 4 bases de datos mostradas para . Se deja para el siguiente report el análisis conjunto de los estos primeros 3 reports para intentar sacar conclusiones utilizando **lógica inductiva** y a partir de esta, decidir qué experimentos realizar a continuación.

Las **conclusiones de este report son muy parecidas a las del anterior**, sólo hemos variado un poco la forma de entrenamiento y establecido en vez de . Resaltamos las algunas de ellas:

* **Valores pequeños de alpha parecen luchar contra el overfitting** produciendo underfitting conforme apilamos capas de una forma tal que el sistema generalice mejor. Mientras que valores grandes suelen mejorar la solución de train (si no hay overtitting también se mejora la de test y todos contentos)
* Las propiedades del sistema para **beta = 0 y beta = 1 son muy similares**. Para bases de datos cuyo alpha óptimo sea bajo merece la pena estudiar este parámetro (Waveform) sin embargo puede que no merezca la pena para otras (Abalone).
* Para las combinaciones óptimas de parámetros (alpha, Nepoch, nH) **el número de capas nL siempre mejora** o simplemente estabiliza el error. El mejor conjunto de parámetros suele ser el que mejor evoluciona con nL (no pasa que un conjunto sea el mejor porque directamente empezó siendo el mejor en las primeras capas).
* El número de **épocas Nepoch** juega un papel importante, el entrenamiento importa. El número óptimo de épocas depende grandemente de alpha (Wav ). Ambos parámetros tienen que ver con el overfitting.
* El número de neuronas óptimo nH depende del problema y del resto de parámetros.
* Las constantes de agregación **gamma** suelen tener una **evolución inicial** corta para luego saturarse en un valor fijo que depende del conjunto de parámetros (alpha, Nepoch, nH). Habría que estudiar **otras formas de agregación**.
* Para los problemas de **kwok y ima** (sobretodo) la evolución del algoritmo es muy pobre en todos los casos (apenas mejora con nL). Hay que seriamente **utilizar otra forma de entrenamiento o explorar más parámetros**.
* Hay que **disminuir los rangos de parámetros que barremos** y hacer un estudio del efecto de los parámetros en los problemas en función de las propiedades de los mismos (se hará en el siguiente report).

## Líneas futuras

De acuerdo a los resultados obtenidos, el siguiente estudio que se realizará se dedicará a estudiar para un conjunto reducido de neuronas y valores de el comportamiento del algoritmo en función de varios parámetros de entrenamiento: , y .

Dado que ya se ha visto que por separado muestran el mismo comportamiento cualitativo, para el siguiente report se va a establecer como valor de compromiso. Se deja para otro report explorar , primero se verá cómo afecta el entrenamiento. El número de capas nL se deja en 200.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BBDD |  |  |  |  |  |
| Abalone | 4,6,8,10 | 1,2,4,8 | 50,100,200 | 8,10,12,14,16 | 0.9,1 |
| Kwok | 1,2,4,6 | 1,2,4,8 | 50,100,200 | 20,25,30,40 | 0.5,1 |
| Image | 4,6,8,10 | 1,2,4,8 | 50,100,200 | 25,30,35,40 | 0.5,1 |
| Waveform | 1,2,4,6 | 1,2,4,8 | 50,100,200 | 2, 3, 4, 5, 8 | 0, 0.1, 0.2 |