



Sistemas Conexionistas y Evolutivos

1^{er} parcial: reporte

— —

19 de febrero de 2016, Puebla.

1. Base de imágenes

Se seleccionaron 42 imágenes de bosques con resoluciones diferentes, 36 de las cuales contienen incendios forestales. Seis mas no contienen ningún signo de fuego, y tres de ellas presentan un bosque en otoño con las hojas de color. Las imágenes se adjuntan en archivo *wildfire.zip*.

Nota

La partición de imágenes en regiones se adapta al tamaño de la imagen, lo que permite recibir regiones con desviaciones de tamaño reducidas. Además, el vector de características se forma por los valores estadísticos de la región, los cuales no están afectados por la diferencia en el tamaño de la muestra.

2. Partición de imágenes

El proceso de partición es parte de la aplicación `image-characteristics`, la cual se discute mas detalladamente en la siguiente sección.

Esta representado por un *clase de tipos* `RegionsExtractor` en el archivo `src/ImgCharacteristics.hs` . Al momento tiene una implementación para `FixedColRowRegions`, el cual describe:

1. el número de filas deseado (el número máximo);
2. el número de columnas deseado (el número máximo);
3. el tamaño mínimo de un *región*.

La implementación se llama `fixedColRowRegions` y se encuentra en `src/ImgCharacteristics/Friday.hs` . Utiliza la función `finalSize` para encontrar el número máximo de columnas y filas, que producirán regiones de tamaño no menor al establecido por `FixedColRowRegions`. La partición de la imagen se hace con la función `crop` de la librería `friday`.

Se espera la declaración de una *instancia* de `RegionsExtractor` en el modulo `Main`, cómo en `exec/DescriptiveStatsAll.hs` (está diseñado de esta manera para evitar posibles conflictos de importación).

3. Vector de características

Un vector de características se extrae de una región de imagen por las clases `CharacteristicExtractor` y `CharacteristicExtractors`, los cuales definen los extractores y sus nombres. Están basados en vectores, indexados con *números naturales en nivel de tipos*, implementados en el proyecto `Nat` .

Los extractores mencionados se construían utilizando `ChanelExtractor` y `LinkedChanelExtractor`, definidos en `src/ImgCharacteristics/ExtractorBuilder.hs` .

Para el problema dado se utilizan los siguientes extractores de canales, definidos en `src/ImgCharacteristics/Friday/Extractors.hs` :

- `mean` — la esperanza matemática;
- `meanQuadratic` — la media cuadrática, $\sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}$;
- `stdev'` — la desviación estándar, requiere valor de la media;
- `min` — el valor mínimo (Q_0);
- `max` — el valor máximo (Q_5);
- `quartiles` — los cuartiles Q_1, Q_2, Q_3 ;

En total son **8** características.

Los extractores de canales se aplican a cada canal de la región (independientemente). Un vector de características tiene dimensión *número de características* \times *número de canales*.

Definición de `ChanelExtractor` utilizado:

```
import ImgCharacteristics.Friday.Extractors as CE

descriptiveStats :: ( NatRules n, Floating num
                    , Ord num, GenVec n
                    , NatRules3Pack n
                    ) => ChanelExtractor n num N8
descriptiveStats = CE.min
                  + # CE.quartiles
                  + # CE.max
                  + # CE.meanQuadratic
                  + ## CE.meanAndVar
```

Para transformar `ChanelExtractor` a `CharacteristicsExtractor` se utilizan las funciones `extractorRGB`, `extractorHSV` y `extractorGrey`.

Se encuentran en `src/ImgCharacteristics/Friday.hs` .

En el ejecutable utilizado (`exec/DescriptiveStatsAll.hs`) se combinan los extractores `RGB`, `HSV`, `Grey` para recibir vectores de dimensión *56* ($8 \times 3 + 8 \times 3 + 8 \times 1$).

4. Datos de aprendizaje

Para el aprendizaje *supervisado*, se requieren las etiquetas de clase para cada instancia de los datos. El funcionamiento de selección de etiquetas está definido

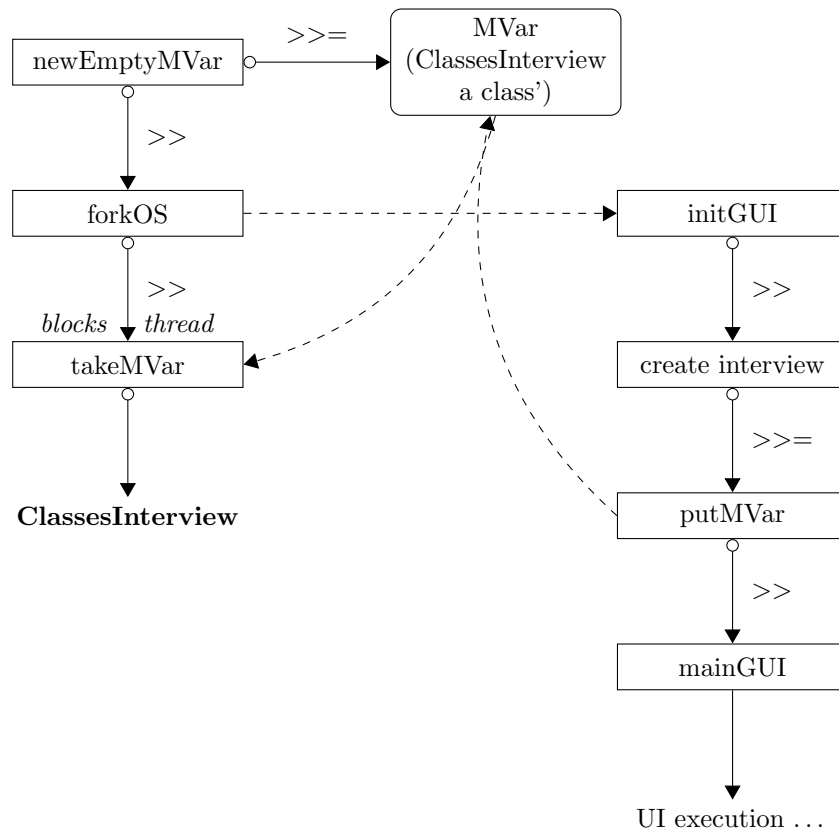
por la clase `RegionsClassesProvider` (en `src/ImgCharacteristics.hs`) y está diseñado para una acción de *entrada/salida*. Provee dos funciones:

1. `classProvider` — crea una instancia de `RegionsClassesProvider`;
2. `regionClass` — pregunta la clase de la imagen (región).

Específicamente, está implementado a través de “GIMP Toolkit” (GTK), el cual se utiliza para la interfaz gráfica “GNOME”. La interfaz de la ventana está proveída por un contenedor `ClassesInterview`, definido en `src/ImgCharacteristics/GTK.hs`, el cual provee:

- `ciWindow` — el objeto de la ventana gráfica;
- `ciAskClass` — una función `a -> IO class'`, donde `a` es una imagen (una región);
- `ciDestroy` — una función para destruir la interfaz gráfica.

Creación del `ClassesInterview`:



Interrogación del usuario `extractLearnData`
 en `src/ImgCharacteristics.hs` y `src/ImgCharacteristics/GTK.hs` :

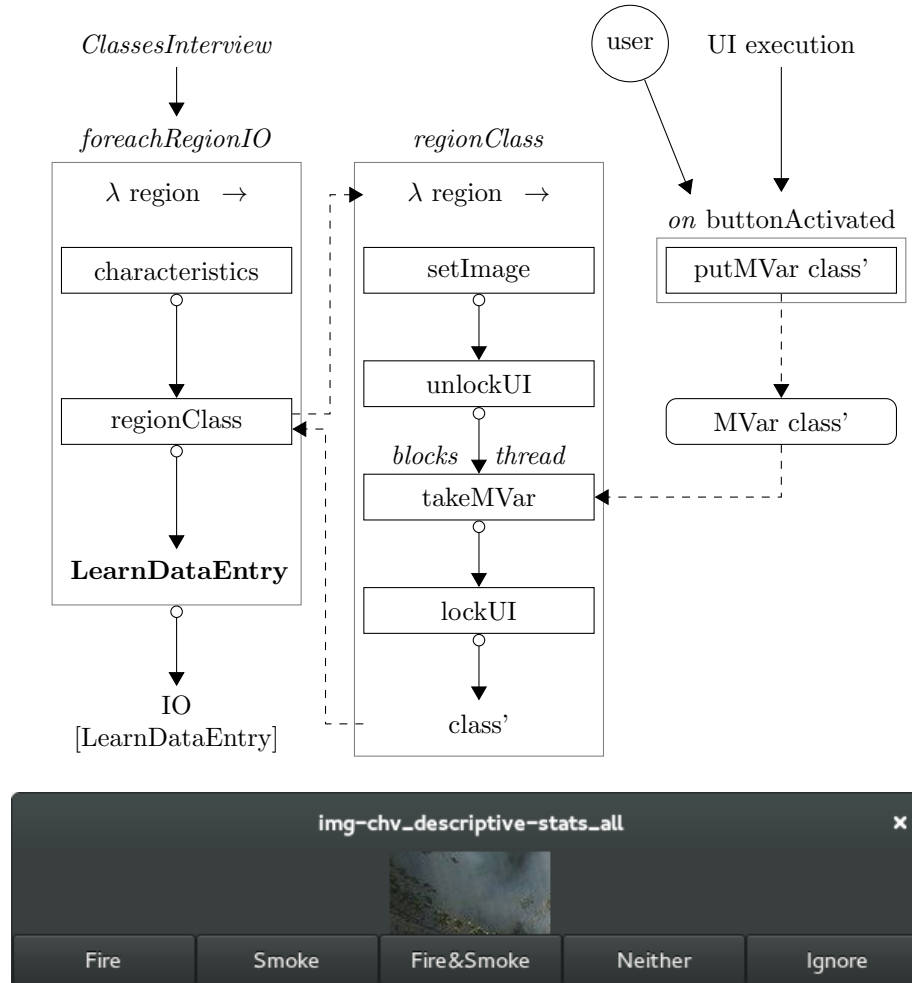


Figura 1: La interfaz gráfica.

4.1. Clases de regiones

Se utilizan 5 clases (`exec/DescriptiveStatsAll.hs`):

1. Fire
2. Smoke
3. Fire&Smoke

4. `Neither`

5. `Ignore`

Las instancias etiquetadas con `Ignore` están a la espera de ser removidos.

4.2. Formato ARFF

Para la escritura de los datos se utiliza el proyecto `WekaData` ; el código fuente se incluye en la carpeta `sources`.

Los adaptadores entre `LearnDataEntry` y `WekaEntry` se encuentran en `src/ImgCharacteristics/Weka.hs` .

4.3. Ejecutable

El proyecto provee varios ejecutables, pero `img-chv_descriptive-stats_all` es el que se usa (`exec/DescriptiveStatsAll.hs`).

Tiene dos modos de uso:

1. `img-chv_descriptive-stats_all images relation target`
2. `img-chv_descriptive-stats_all --dir directory relation target`
 - images* — enumeración de imágenes;
 - directory* — un directorio con imágenes a procesar;
 - relation* — nombre de la relación (`@relation`);
 - target* — el archivo **.arff* a escribir.

4.4. Compilación

Requiere:

- The Glorious Glasgow Haskell Compilation System (GHC), version 7.10.3.
- Common Architecture for Building Applications and Libraries (CABAL), version ≥ 1.10 .
- *ghc-gtk3*, version ≥ 0.14 (puede ser complicada a compilar).
- Proyecto *Nat*.
- Proyecto *WekaData*.

Las demás dependencias deben ser resueltas por CABAL sin problemas.

5. Experimentos en Weka

Para automatizar el proceso de aprendizaje del perceptron, fue creado un pequeño proyecto en Scala `WekaANN`. Se aprovecha la facilidad de conexión de aplicaciones de JMV para correr las *evaluaciones* de Weka.

- Prepara los datos, removiendo las instancias marcadas con la clase `Ignore`.
- Guarda los reportes de Weka en archivos (por defecto en directorio “reports”).

En el directorio del proyecto *scripts* se encuentran estos programas:

1. `bulkANN.scala` — fue utilizado para prueba de configuraciones diferentes.
2. `findBestReports.scala` — selecciona los mejores reportes (por el porcentaje de clasificación correcta).
3. `extractReportsInfo.scala` — extrae estadísticas desde los reportes y las escribe en un archivo.

En el directorio `octave` se encuentran diferentes utilidades para graficar los comparaciones de los resultados (para Matlab/Octave).

5.1. Configuraciones de la red

Fueron realizados experimentos con todas las combinaciones de los siguientes valores:

Variable	Valores
Número de épocas	500, 1000, 2000, 3000
“Learning rate”	0.1, 0.2, 0.5, 0.8
“Momentum”	0.2, 0.5, 0.8
1 ^{ra} capa oculta	a, t
2 ^{da} capa oculta	0, a, t

Total: 288 combinaciones.

$$t = \text{núm. de classes} + \text{núm. de atributos} = 56 + 5 = 61; a = \frac{t}{2} = 30.$$

6. Análisis de resultados

Para analizar la influencia de diferentes variables sobre la calidad de aprendizaje, fueron graficados unas proyecciones, los cuales parecen a una histograma, de siguiente manera:

#	épocas	learn.	mom.	capa 1	capa 2	clasificado	rel. abs. err.
1	500	0.1	0.5	t	a	78.3044 %	32.4691 %
2	500	0.1	0.5	t	t	78.1464 %	32.8638 %
3	3000	0.1	0.2	t	a	78.0411 %	32.1325 %
4	3000	0.1	0.2	t		77.8831 %	34.6127 %
5	1000	0.1	0.8	t	t	77.8304 %	33.1316 %
6	1000	0.1	0.2	t	a	77.7251 %	33.4992 %
7	1000	0.1	0.2	t		77.7251 %	35.4322 %
8	3000	0.1	0.5	t		77.6198 %	34.8487 %
9	1000	0.2	0.8	t	t	77.6198 %	33.7434 %
10	2000	0.5	0.2	t	t	77.6198 %	33.8068 %

Tabla 1: Las configuraciones con los mejores resultados.

#	épocas	learn.	mom.	capa 1	capa 2	clasificado	rel. abs. err.
1	3000	0.8	0.8	a	t	11.743 %	125.7089 %
2	2000	0.8	0.8	t	t	15.3765 %	120.8736 %
3	1000	0.8	0.8	t	t	16.7983 %	120.174 %
4	3000	0.8	0.8	t	t	20.5898 %	115.5593 %
5	2000	0.8	0.8	a	t	22.9595 %	111.3145 %

Tabla 2: Las configuraciones con los peores resultados.

Sean $C = c_i$ — variables configurables.

$E = e_i$ — estadísticas para las configuraciones.

$D = d_j$ — arreglos de configuraciones, tales que

$\forall d_j \in D$

$\forall c_i \in C \exists$ un valor d_{ij}^c asociado — el valor de la variable c_i
en la configuración d_j ;

$\forall e_i \in E \exists$ un valor d_{ij}^e asociado — el valor de la estadística e_i
en la configuración d_j .

Una proyección de variables $X, Y \in C$ sobre la estadística $Z \in E$ con operador op es el conjunto de valores $\{p_{xy}\}$, tales que

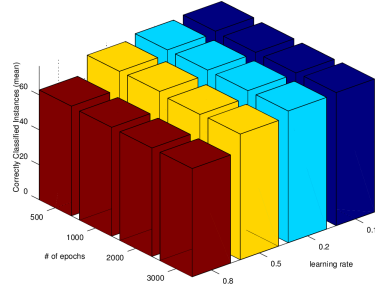
\forall único $x \in$ valores de X

\forall único $y \in$ valores de Y

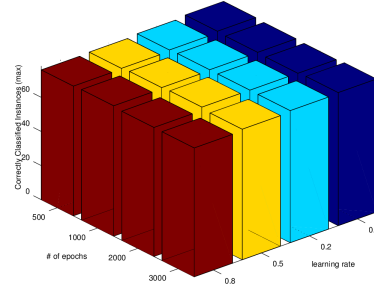
$p_{xy} = op(\{z_{xy}\})$, donde

$\{z_{xy}\} = \{d_{Zj}^e \mid \forall d_j \in D \text{ si } d_{Xk}^c = x \wedge d_{Yk}^c = y\}$

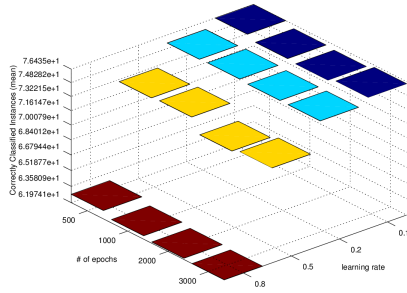
El operador op puede ser el *promedio*, o valor *máximo/mínimo*, u otra función con tipo $\{z\} \mapsto z$.



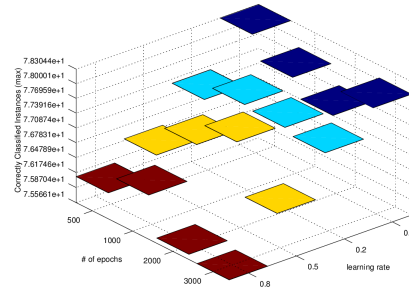
(a) Escala lineal.



(b) Escala lineal.



(c) Escala logarítmica.



(d) Escala logarítmica.

Figura 2: La proyección del *número de épocas* utilizado y el “*learning rate*” sobre la **media** (izquierda: 2a, 2c) y el **máximo** (derecha: 2b, 2d) del *número de instancias correctamente clasificadas*.

Se puede ver que los gráficos en la figura 5 son una reflexión de los gráficos en la figura 4.

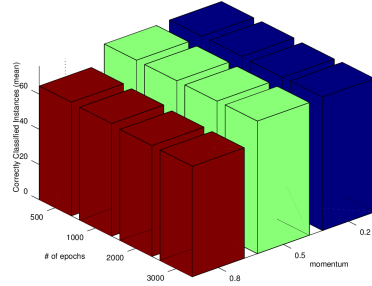
Pueden hacerse las siguientes conclusiones:

6.1. Influencia de “learning rate”

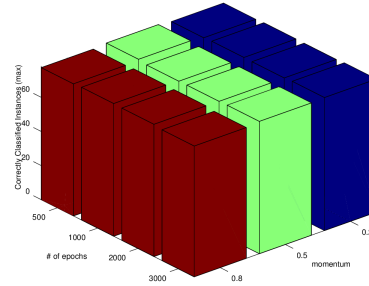
La calidad disminuye con el aumento del “learning rate” (figura 2). Las 5 configuraciones, que mostraron los peores resultados (tabla 2), todos tenían el l.r. máximo de las pruebas (0.8).

6.2. Influencia de “momentum”

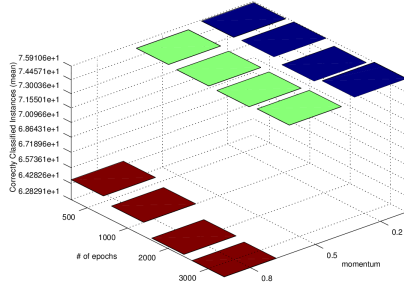
En la figura 3 se puede ver que el mayor valor (0.8) del “momentum” produjo los peores resultados. La diferencia de los promedios para 0.5 y 0.2 está



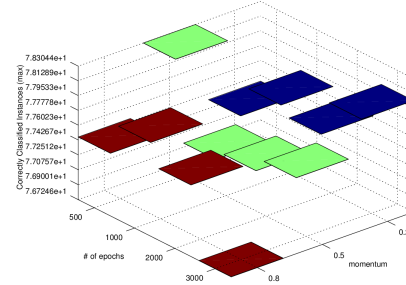
(a) Escala lineal.



(b) Escala lineal.



(c) Escala logarítmica.



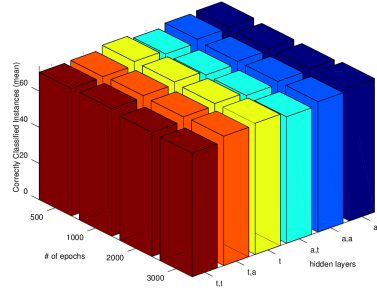
(d) Escala logarítmica.

Figura 3: La proyección del *número de épocas* utilizado y el “*momentum*” sobre la **media** (izquierda: 3a, 3c) y el **máximo** (derecha: 3b, 3d) del *número de instancias correctamente clasificadas*.

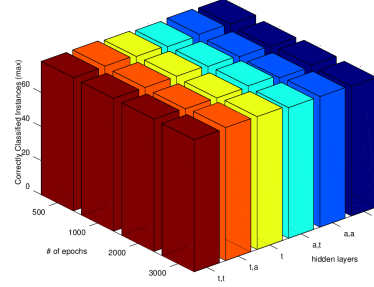
ligeramente en favor del valor menor, mientras que los máximos son inconclusos.

6.3. Influencia de la capa oculta

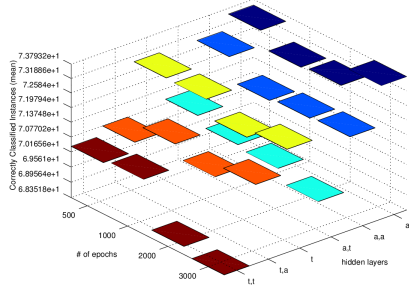
- a). Las 10 mejores configuraciones tenían $t = \frac{a}{2} = 61$ neuronas en la primera capa. En las figuras 4, 5 y tabla 1 se puede ver que
 - los mejores resultados (los máximos) fueron producidos por las redes de dos capas: $t, a = 61, 30$ y $t, t = 61, 61$.
 - la configuración con el resultado siguiente fue de una sola capa $t = 61$.
 - los peores máximos fueron dados por la configuración $a = 30$.
- b). Pero en las figuras 4 y 5 se puede ver que en promedio, los mejores resultados fueron dados por la configuración de una capa: $a = 30$. Esto significa que está menos influenciada por los demás parámetros. Igual, las redes t, a y t, t mostraron promedios notablemente peores que sus máximos.



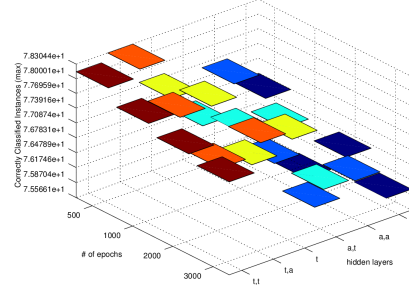
(a) Escala lineal.



(b) Escala lineal.



(c) Escala logarítmica.



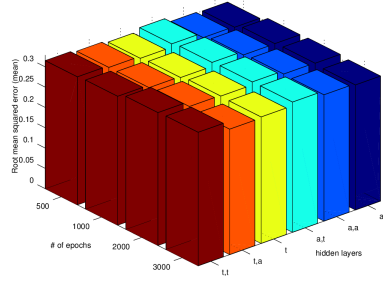
(d) Escala logarítmica.

Figura 4: La proyección del *número de épocas* utilizado y la *configuración de las capas escondidas* sobre la **media** (izquierda: 4a, 4c) y el **máximo** (derecha: 4b, 4d) del *número de instancias correctamente clasificadas*.

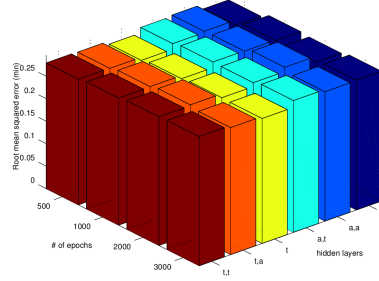
c). La configuración $a, t = 30, 61$ no mostró ni bien máximo, ni bien promedio.

6.4. Matrices de confusión

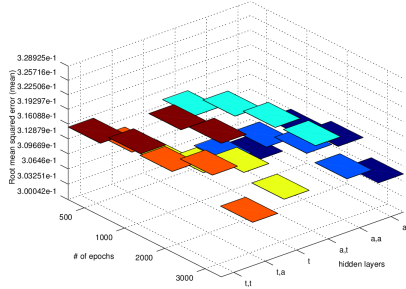
1. **Fire&Smoke** mostró el peor ratio de clasificación *verdadera positiva*.
2. **Neither** mostró el mejor ratio de clasificación *verdadera positiva*, después sigue **Smoke**.
3. **Neither** también demuestra el mayor ratio de clasificación *falsa positiva*.
4. El menor ratio de clasificación *falsa positiva* fue mostrado por clases **Fire** y **Fire&Smoke**.
5. Aun que la clase **Neither** mostró el peor ratio de falsas positivas, demuestra también la mejor precisión entre las clases.
6. **Fire&Smoke** tiene la peor precisión de las clases utilizadas.



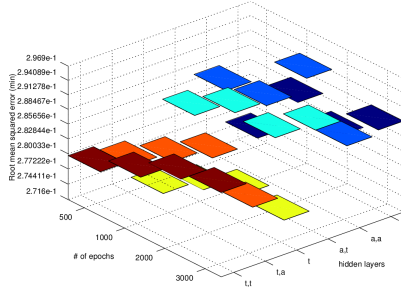
(a) Escala lineal.



(b) Escala lineal.



(c) Escala logarítmica.



(d) Escala logarítmica.

Figura 5: La proyección del *número de épocas* utilizado y la *configuración de las capas escondidas* sobre la **media** (izquierda: 5a, 5c) y el **mínimo** (derecha: 5b, 5d) de la *error cuadrática media*.

Cómo se puede ver en la figura 7:

- Las clases **Fire** y **Smoke** se distinguen bien.
- Se distingue bien **Neither** de **Fire**, pero no al contrario.
- La clase **Fire&Smoke** se distingue mal de ambos **Fire** y **Smoke**.
- **Fire** se distingue mal de **Fire&Smoke**.
- **Fire** y **Smoke** tienen en porcentaje de equivocaciones en favor de **Neither** parecido ($\approx 14\%$).

(1)

a	b	c	d	e	← classified as
200	8	49	41	0	a = Fire
12	407	29	70	0	b = Smoke
54	41	132	22	0	c = Fire&Smoke
28	51	7	748	0	d = Neither
0	0	0	0	0	e = Ignore

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.671	0.059	0.68	0.671	0.676	0.901	Fire
	0.786	0.072	0.803	0.786	0.794	0.928	Smoke
	0.53	0.052	0.608	0.53	0.567	0.869	Fire&Smoke
	0.897	0.125	0.849	0.897	0.872	0.931	Neither
	0	0	0	0	0	?	Ignore
Weighted Avg.	0.783	0.091	0.778	0.783	0.78	0.917	

(2)

a	b	c	d	e	← classified as
212	7	46	33	0	a = Fire
6	413	35	64	0	b = Smoke
62	50	121	16	0	c = Fire&Smoke
38	33	25	738	0	d = Neither
0	0	0	0	0	e = Ignore

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.711	0.066	0.667	0.711	0.688	0.919	Fire
	0.797	0.065	0.821	0.797	0.809	0.921	Smoke
	0.486	0.064	0.533	0.486	0.508	0.865	Fire&Smoke
	0.885	0.106	0.867	0.885	0.876	0.945	Neither
	0	0	0	0	0	?	Ignore
Weighted Avg.	0.781	0.083	0.779	0.781	0.78	0.924	

(3)

a	b	c	d	e	← classified as
201	13	41	43	0	a = Fire
7	417	46	48	0	b = Smoke
54	48	130	17	0	c = Fire&Smoke
31	49	20	734	0	d = Neither
0	0	0	0	0	e = Ignore

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.674	0.057	0.686	0.674	0.68	0.91	Fire
	0.805	0.08	0.791	0.805	0.798	0.93	Smoke
	0.522	0.065	0.549	0.522	0.535	0.87	Fire&Smoke
	0.88	0.101	0.872	0.88	0.876	0.938	Neither
	0	0	0	0	0	?	Ignore
Weighted Avg.	0.78	0.084	0.778	0.78	0.779	0.922	

6.5. Notas de mejoramiento

1. Configuración de la red:

- Utilizar “learning rate” en el rango $(0, 0.2]$.
- Ajustar “momentum” ≤ 0.5 .
- Usar una sola capa con t neuronas, o de dos capas, con t neuronas en la primera capa.
- Investigar el mejor tiempo de entrenamiento para las configuraciones mencionadas.

2. Metodología:

- Remover clase **Fire&Smoke**, porque no solamente su calidad de clasificación es mala, pero todavía atrae alrededor de 16 % de instancias de clase **Fire**.
- Establecer la “cantidad” mínima de fuego/humo en la región de imagen para considerarla de la clase correspondiente; ignorar los regiones que no tienen bastante de evidencia. Eso debería de mejorar la calidad de distinción para **Fire** \rightarrow **Neither** y **Smoke** \rightarrow **Neither**.

(4)

	a	b	c	d	e	← classified as
190	12	55	41	0	0	a = Fire
3	421	35	59	0	0	b = Smoke
50	40	138	21	0	0	c = Fire&Smoke
33	54	17	730	0	0	d = Neither
0	0	0	0	0	0	e = Ignore

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.638	0.054	0.688	0.638	0.662	0.913	Fire
	0.813	0.077	0.799	0.813	0.806	0.942	Smoke
	0.554	0.065	0.563	0.554	0.559	0.866	Fire&Smoke
	0.875	0.114	0.858	0.875	0.866	0.942	Neither
	0	0	0	0	0	?	Ignore
Weighted Avg.	0.779	0.088	0.777	0.779	0.777	0.928	

(5)

	a	b	c	d	e	← classified as
204	7	48	39	0	0	a = Fire
5	410	37	66	0	0	b = Smoke
56	46	126	21	0	0	c = Fire&Smoke
34	49	13	738	0	0	d = Neither
0	0	0	0	0	0	e = Ignore

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.685	0.059	0.682	0.685	0.683	0.907	Fire
	0.792	0.074	0.801	0.792	0.796	0.933	Smoke
	0.506	0.059	0.563	0.506	0.533	0.838	Fire&Smoke
	0.885	0.118	0.854	0.885	0.869	0.941	Neither
	0	0	0	0	0	?	Ignore
Weighted Avg.	0.778	0.089	0.774	0.778	0.776	0.92	

(6)

	a	b	c	d	e	← classified as
197	6	58	37	0	0	a = Fire
10	400	38	70	0	0	b = Smoke
50	44	135	20	0	0	c = Fire&Smoke
34	43	13	744	0	0	d = Neither
0	0	0	0	0	0	e = Ignore

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.661	0.059	0.677	0.661	0.669	0.909	Fire
	0.772	0.067	0.811	0.772	0.791	0.929	Smoke
	0.542	0.066	0.553	0.542	0.548	0.867	Fire&Smoke
	0.892	0.119	0.854	0.892	0.873	0.941	Neither
	0	0	0	0	0	?	Ignore
Weighted Avg.	0.777	0.089	0.775	0.777	0.776	0.923	

Figura 6: Matrices de confusión para los mejores resultados 1-6.

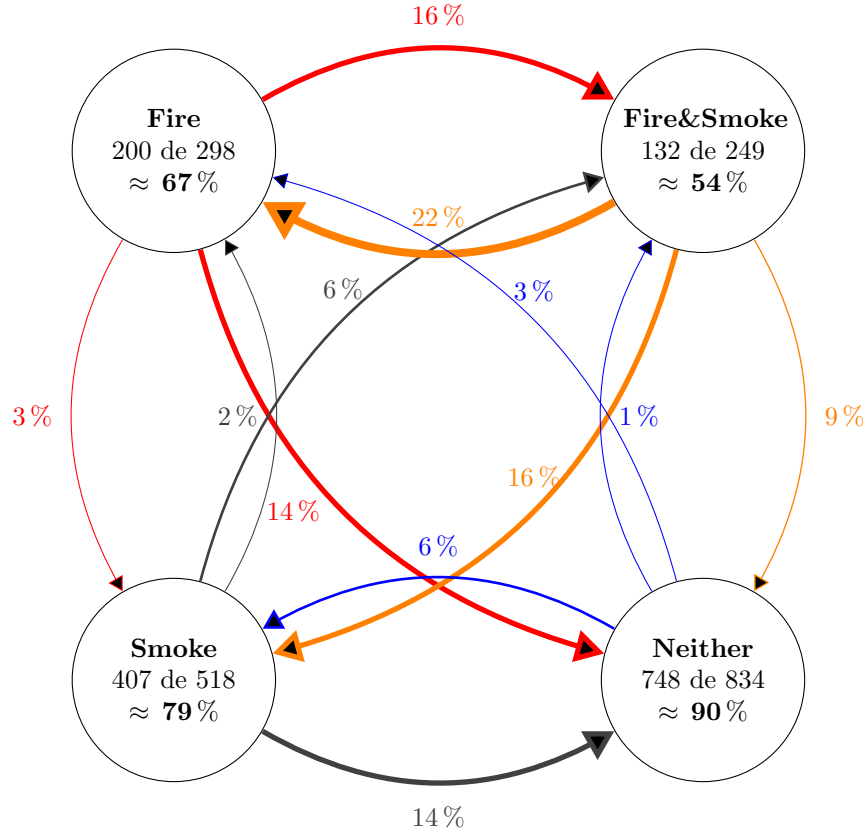


Figura 7: Diagrama de clasificación para #1 de la tabla 1. En los nodos se presenta la clase y el ratio de clasificación *verdadera positiva*. Las flechas significan la clasificación equivocada y apuntan a las clases que fueron asignadas por error. Están etiquetadas por el ratio de clasificación *falsa positiva* y el grosor de la línea también lo refleja.