Información mutua y métodos contrastivos en el aprendizaje de representaciones

Doble Grado en Ingeniería Informática y Matemáticas

Francisco Javier Sáez Maldonado

13 de septiembre de 2021

Trabajo Fin de Grado

UNIVERSIDAD
DE GRANADA

E.T.S. de Ingenierías Informática y de Telecomunicación
Facultad de Ciencias

Índice

Motivación

Dato	Etiqueta
$(0.1, 0, 2, 1, 0, 0.5, 2.4, 5)$	Perro

Motivación

Dato	Etiqueta
$(0.1, 0, 2, 1, 0, 0.5, 2.4, 5)$	Perro

Sea $x \in \mathbb{R}^d$ un vector de entrada a un modelo de aprendizaje automático. Una *representación* $\tilde{x} \in \mathbb{R}^n$ es otro vector de menor dimensión que comparte información o características con x.

Objetivo: extraer **representaciones** que sean buenas en general para **tareas posteriores**.

Teoría de la información

Divergencia Kullback-Leibler

Definición (Divergencia Kullback-Leibler)

Sean P y Q dos distribuciones de probabilidad sobre el mismo espacio probabilístico, su divergencia de Kullback-Leibler $D_{KL}\Big(Q \,||\, P\Big)$ mide la "diferencia" de Q a P

$$D_{KL}\Big(P \,||\, Q\Big) = E_P \log \frac{P(x)}{Q(x)}.$$

La divergencia de Kullback-Leibler es siempre no negativa.

Definición (Información mutua)

Sean X, Z variables aleatorias. La información mutua entre ellas se expresa como

$$I(X, Z) = H(X) - H(X \mid Z).$$

También se puede expresar como

$$I(X, Z) = D_{KL}\Big(P_{XZ} \,||\, P_X P_Z\Big).$$

Cotas inferiores de la información mutua

Proposición (Cota inferior variacional)

Sean X, Z variables aleatorias y $Q_\theta(Z \mid X)$ una distribución de probabilidad arbitraria. Entonces,

$$I(X, Z) \geq H(Z) + E_{P_X}\left[E_{P_{Z|X}}\left[\log Q_\theta(Z \mid X)\right]\right].$$

En el contexto del aprendizaje automático, podemos considerar que Q_θ es una red neuronal y maximizar la cota inferior usando un algoritmo de optimización como *backpropagation*.

Teorema (Representación Donsker-Varadhan)

La divergencia de Kullback-Leibler entre las distribuciones P y Q también puede expresarse como

$$D_{KL}(P \,||\, Q) = \sup_{T} E_P[T] - \log E_Q\left[e^T\right],$$

donde el supremo se toma sobre todas las funciones $T : \Omega \to \mathbb{R}$ que hacen que la esperanza bajo P exista.

Teorema (Representación Donsker-Varadhan)

La divergencia de Kullback-Leibler entre las distribuciones P y Q también puede expresarse como

$$D_{KL}(P \mid\mid Q) = \sup_{T} E_P[T] - \log E_Q\left[e^T\right],$$

donde el supremo se toma sobre todas las funciones $T : \Omega \to \mathbb{R}$ que hacen que la esperanza bajo P exista.

Corolario

Sea $\mathcal{F}$ una clase de funciones $T : \Omega \to \mathbb{R}$ que satisfacen las condiciones del teorema anterior. Entonces:

$$I(P, Q) = D_{KL}\Big(P \mid\mid Q\Big) \geq \sup_{T \in \mathcal{F}} E_P[T] - \log E_Q\left[e^T\right].$$

Aprendizaje contrastivo

Estimación del ruido contrastiva - Problema

Consideramos:

- $X = \{x_1, \ldots, x_{T_d}\}$ una muestra que suponemos extraída de una distribución $P_d \in \{P_m(.;\theta)\}_\theta$.
- $Y = \{y_1, \ldots, y_{T_n}\}$ una muestra de elementos idénticamente distribuidos, que asumimos extraída de una distribución de ruido conocida P_n.

Estimación del ruido contrastiva - Problema

Consideramos:

- $X = \{x_1, \ldots, x_{T_d}\}$ una muestra que suponemos extraída de una distribución $P_d \in \{P_m(.;\theta)\}_\theta$.
- $Y = \{y_1, \ldots, y_{T_n}\}$ una muestra de elementos idénticamente distribuidos, que asumimos extraída de una distribución de ruido conocida P_n.

Problema: Considerando el conjunto $U = X \cup Y = \{u_1, \ldots, u_{T_d+T_n}\}$, ser capaces de discriminar entre elementos de U que fueron extraídos de P_d y elementos extraídos de P_n.

Estimación del ruido contrastiva - Problema

Consideramos:

- $X = \{x_1, \ldots, x_{T_d}\}$ una muestra que suponemos extraída de una distribución $P_d \in \{P_m(.;\theta)\}_\theta$.
- $Y = \{y_1, \ldots, y_{T_n}\}$ una muestra de elementos idénticamente distribuidos, que asumimos extraída de una distribución de ruido conocida P_n.

Problema: Considerando el conjunto $U = X \cup Y = \{u_1, \ldots, u_{T_d+T_n}\}$, ser capaces de discriminar entre elementos de U que fueron extraídos de P_d y elementos extraídos de P_n.

Trataremos de estimar el ratio P_d/P_n y usaremos este ratio para conocer propiedades sobre la distribución P_d.

Estimación del ruido contrastiva - Resolución

Asignando a cada elemento de U una *etiqueta* para poder aplicar la regresión logística

$$C_t(u_t) = \begin{cases} 1 & si\ u_t \in X \\ 0 & si\ u_t \in Y \end{cases} \implies \begin{cases} P(u \mid C = 1, \theta) = P_m(u; \theta) \\ P(u \mid C = 0) = P_n(u) \end{cases}$$

Llamaremos $G(u; \theta)$ al logaritmo del ratio que queremos estimar

$$G(u; \theta) = \log \frac{P_m(u; \theta)}{P_n(u)}.$$

Estimación del ruido contrastiva - Resolución

Proposición

En las condiciones presentadas y llamando $h(u;\theta) := P(C = 1|u;\theta)$*, se tiene que*

$$h(u;\theta) = r_\nu(G(u;\theta)), \quad \textit{donde} \quad r_\nu(u) = \frac{1}{1 + \nu exp(-u)}$$

es la función logística parametrizada por $\nu = P(C = 0)/P(C = 1)$.

Estimación del ruido contrastiva - Resolución

Proposición

En las condiciones presentadas y llamando $h(u;\theta) := P(C = 1|u;\theta)$*, se tiene que*

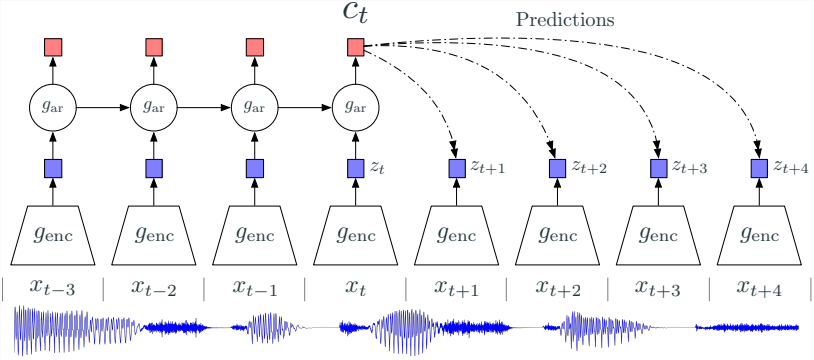
$$h(u;\theta) = r_\nu(G(u;\theta)), \quad \textit{donde} \quad r_\nu(u) = \frac{1}{1 + \nu exp(-u)}$$

es la función logística parametrizada por $\nu = P(C = 0)/P(C = 1)$.

Puesto que las etiquetas C_t siguen una distribución de Bernoulli y son independientes, el logaritmo de la verosimilitud condicionada tiene la forma

$$\ell(\theta) = \sum_{t=1}^{T_d} \log[h(x_t;\theta)] + \sum_{t=1}^{T_n} \log[1 - h(y_t,\theta)]$$

Contrastive predictive coding



- x_{t_j} es la entrada a la red en el instante de tiempo t_j.
- g_{enc} es un codificador que produce una representación $g_{enc}(x_t) = z_t$
- g_{ar} es un modelo autorregresivo que produce una representación que tiene en cuenta el contexto $g_{ar}(z_{\leq t}) = c_t$

Pérdida contrastiva y cota inferior contrastiva

Definición (Pérdida contrastiva)

Sea $X = \{x^*, x_1, \ldots, x_{N-1}\}$ *un conjunto de* N *ejemplos donde* x^* *ha sido extraido de la distribución conjunta* $P(x, z)$ *y le resto han sido extraídos del producto de las distribuciones marginales* $P(x), P(z)$*. Se define entonces la función de pérdida contrastiva como*

$$\ell(\theta) = -E_X \left[\log \frac{h_\theta(x^*, z)}{\sum_{x \in X} h_\theta(x, z)} \right].$$

Pérdida contrastiva y cota inferior contrastiva

Definición (Pérdida contrastiva)

Sea $X = \{x^*, x_1, \ldots, x_{N-1}\}$ un conjunto de N ejemplos donde x^* ha sido extraido de la distribución conjunta $P(x, z)$ y le resto han sido extraídos del producto de las distribuciones marginales $P(x), P(z)$. Se define entonces la función de pérdida contrastiva como

$$\ell(\theta) = -E_X \left[\log \frac{h_\theta(x^*, z)}{\sum_{x \in X} h_\theta(x, z)} \right].$$

Proposición

En las mismas condiciones que en la definición anterior, se tiene que

$$I(x^*, z) \geq -\ell(\theta) + \log N.$$

Funciones de pérdida usando tripletas

Original x

Ejemplo positivo x^+

Ejemplo negativo x^-

Nos interesa tener un margen $\alpha \in \mathbb{R}$

$$\left\|g(x) - g(x^+)\right\|_2 + \alpha < \left\|g(x) - g(x^-)\right\|_2,$$

lo que nos lleva a la pérdida de una tripleta individual

$$\ell^\alpha(x, x^+, x^-) = \text{máx}\left(0, \left\|g(x) - g(x^+)\right\|_2^2 - \left\|g(x) - g(x^-)\right\|_2^2 + \alpha\right).$$

Funciones de pérdida usando tripletas

Definición

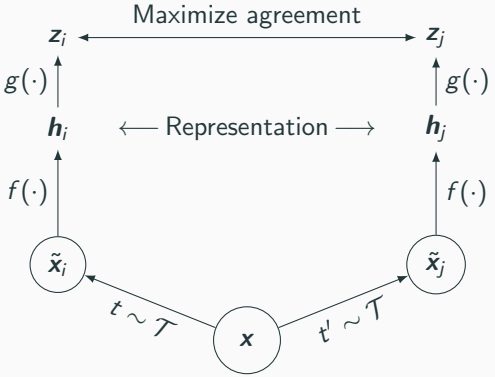
Dado un conjunto de tripletas, cada una con una imagen original, un ejemplo positivo y uno negativo $\mathcal{T} = \{(x_i, x_i^+, x_i^-)\}_{i \in \Lambda}$*, una función de pérdida usando tripletas se define como:*

$$\mathcal{L}(x_i, x_i^+, x_i^-) = \sum_{i \in \Lambda} \ell^\alpha(x_i, x_i^+, x_i^-).$$

Podemos generalizar esta función de pérdida a un caso más eficiente y, a continuación, se puede probar que se está generalizando la pérdida contrastiva.

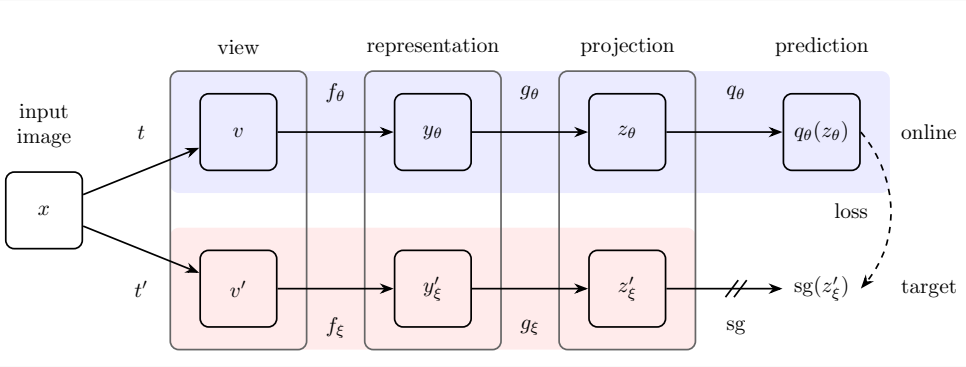
Nuevos marcos de trabajo

SimCLR



$$\ell_{i,j} = -\log \frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} I_{k \neq i} exp(sim(z_i, z_k)/\tau)},$$

Bootstrap your own latent



$$\mathcal{L}_{\theta,\xi} = \left\| \overline{q_\theta}(z_\theta) - \overline{z'_\xi} \right\|_2^2$$

$$\xi \leftarrow \tau\xi + (1-\tau)\theta$$

Experimentos

Problemas y objetivos

Problemas:

Problemas y objetivos

Problemas:

- El conjunto de datos usado en los artículos originales es demasiado grande

Problemas y objetivos

Problemas:

- El conjunto de datos usado en los artículos originales es demasiado grande
- La capacidad de la que disponemos es claramente inferior (1GPU vs 128 TPU)

Problemas y objetivos

Problemas:

- El conjunto de datos usado en los artículos originales es demasiado grande
- La capacidad de la que disponemos es claramente inferior (1GPU vs 128 TPU)

Objetivos:

- Adaptar los experimentos a un conjunto de datos más pequeño

Problemas y objetivos

Problemas:

- El conjunto de datos usado en los artículos originales es demasiado grande
- La capacidad de la que disponemos es claramente inferior (1GPU vs 128 TPU)

Objetivos:

- Adaptar los experimentos a un conjunto de datos más pequeño
- Experimentar con los hiperparámetros de los modelos para comprobar si las hipótesis originales se siguen cumpliendo

Problemas y objetivos

Problemas:

- El conjunto de datos usado en los artículos originales es demasiado grande
- La capacidad de la que disponemos es claramente inferior (1GPU vs 128 TPU)

Objetivos:

- Adaptar los experimentos a un conjunto de datos más pequeño
- Experimentar con los hiperparámetros de los modelos para comprobar si las hipótesis originales se siguen cumpliendo

Tecnologías (Python):

- Tensorflow
- Jax
- Tensorboard

Primer experimento SimCLR

Primer acercamiento, búsqueda en malla para explorar el comportamiento de los hiperparámetros:

Primer experimento SimCLR

Primer acercamiento, búsqueda en malla para explorar el comportamiento de los hiperparámetros:

- Tamaño de batch
- Temperatura τ
- Intensidad de la fluctuación de color

Primer experimento SimCLR

Primer acercamiento, búsqueda en malla para explorar el comportamiento de los hiperparámetros:

- Tamaño de batch
- Temperatura τ
- Intensidad de la fluctuación de color

Este primer experimento se hace utilizando **ResNet18** como codificador.

Primer experimento SimCLR

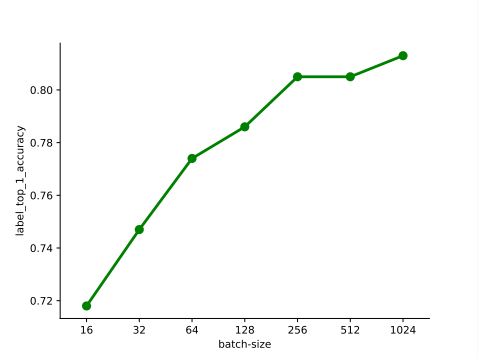
Primer acercamiento, búsqueda en malla para explorar el comportamiento de los hiperparámetros:

- Tamaño de batch
- Temperatura τ
- Intensidad de la fluctuación de color

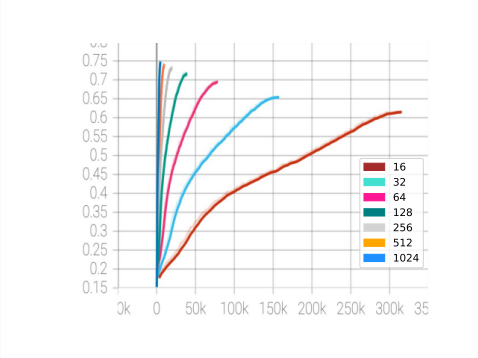
Este primer experimento se hace utilizando **ResNet18** como codificador.

Procedencia	batch_size	temperature	color_jitter	regularization_loss	top_1_accuracy	top_5_accuracy
Propio	512	0.25	0.25	0.0093	0.833	0.994
Propio	1024	0.25	0.75	0.0093	0.841	0.995
Original	512	0.5	-	-	~ 0.846	-
Original	1024	0.5	-	-	~ 0.851	-

Influencia del tamaño de batch



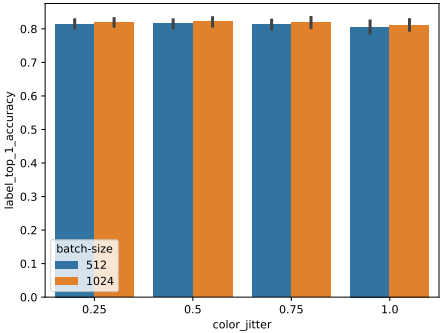
Evolución de la precisión según el tamaño del batch.



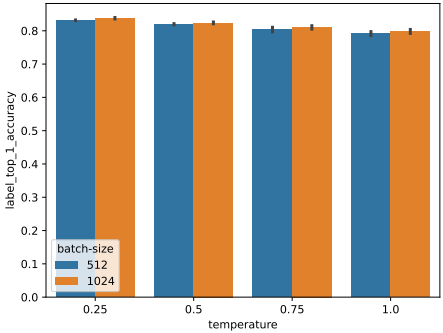
Curvas de evolución de la precisión en función del tamaño del batch.

En la evaluación de la pérdida, se compara con el resto del batch.

Influencias de la intensidad de fluctuación del color y de la temperatura



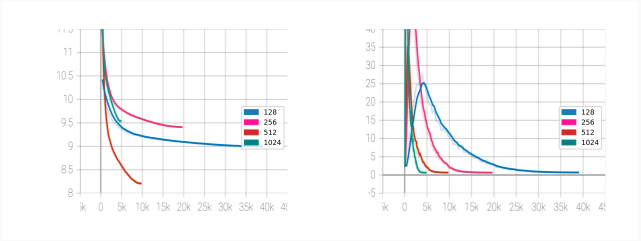
Influencia de la intensidad de fluctuación.



Influencia de la temperatura.

Segundo experimento SimCLR

Aumentamos la profundidad del codificador, usamos **ResNet50**.



Pérdida contrastiva en train Pérdida supervisada

resnet_depth	batch_size	temperature	color_jitter	regularization_loss	top_1_accuracy	top_5_accuracy	steps
18	1024	0.25	0.75	0.0093	0.841	0.995	4900
50	128	0.5	0.65	0.0225	0.844	0.994	39100
	256	**0.5**	**0.75**	**0.023**	**0.848**	**0.994**	**19695**

Tercer experimento y resultados finales

Añadimos **alisado gaussiano** al preprocesamiento.

Experimento	resnet_depth	batch_size	temperature	color_jitter	regularization_loss	label_top_1_accuracy	label_top_5_accuracy	global_step
1	18	1024	0.25	0.75	0.0093	0.841	0.995	4900
2	50	256	0.5	0.75	0.023	0.848	994	19695
3	18	1024	0.25	0.65	0.0093	0.846	0.994	4900
	50	**256**	**0.25**	**0.65**	**0.0228**	**0.862**	**0.997**	**19695**
Original	50	512	0.5	-	-	$\sim$ 0.846	-	9800
	50	1024	0.5	-	-	$\sim$ 0.851	-	4900

Tercer experimento y resultados finales

Añadimos **alisado gaussiano** al preprocesamiento.

Experimento	resnet_depth	batch_size	temperature	color_jitter	regularization_loss	label_top_1_accuracy	label_top_5_accuracy	global_step
1	18	1024	0.25	0.75	0.0093	0.841	0.995	4900
2	50	256	0.5	0.75	0.023	0.848	994	19695
3	18	1024	0.25	0.65	0.0093	0.846	0.994	4900
	50	**256**	**0.25**	**0.65**	**0.0228**	**0.862**	**0.997**	**19695**
Original	50	512	0.5	-	-	$\sim$ 0.846	-	9800
	50	1024	0.5	-	-	$\sim$ 0.851	-	4900

Encontramos un **bug** en el código de Google a la hora de realizar la transferencia de aprendizaje a otros dataset.

Experimento con BYOL

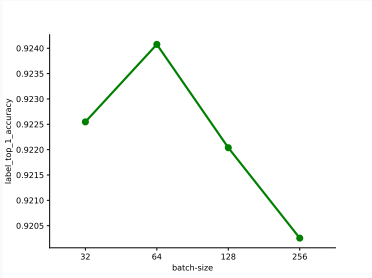
Se pretende comprobar (usando **ResNet50**) si el tamaño de batch sigue siendo relevante y si se obtienen mejores resultados que los obtenidos en SimCLR.

$$\mathcal{L}_{\theta,\xi} = \left\| \overline{q_\theta}(z_\theta) - \overline{z'_\xi} \right\|_2^2$$

Experimento con BYOL

Se pretende comprobar (usando **ResNet50**) si el tamaño de batch sigue siendo relevante y si se obtienen mejores resultados que los obtenidos en SimCLR.

$$\mathcal{L}_{\theta,\xi} = \left\| \overline{q_\theta}(z_\theta) - \overline{z'_\xi} \right\|_2^2$$



batch_size	loss	top_1_accuracy	top_5_accuracy	steps
64	**0.381**	**0.9240764**	**1**	**148000**

Conclusiones

- El uso de la teoría de la información proporciona un buen punto de partida para el aprendizaje de representaciones.
- El aprendizaje contrastivo ha probado ser la mejor forma de obtener representaciones que son útiles en tareas posteriores.
- Ambos marcos de trabajo probados obtienen buenos resultados en la adaptación a conjuntos de datos más pequeños.

Gracias por su atención

Nota final

El contenido que se expondrá en esta presentación está explicado detalladamente en un documento en

`https://github.com/fjsaezm/Mutual-Information-in-Unsupervised-Machine-Learning`.

El código que se ha elaborado para ayudarnos en este trabajo también se encuentra en este repositorio.