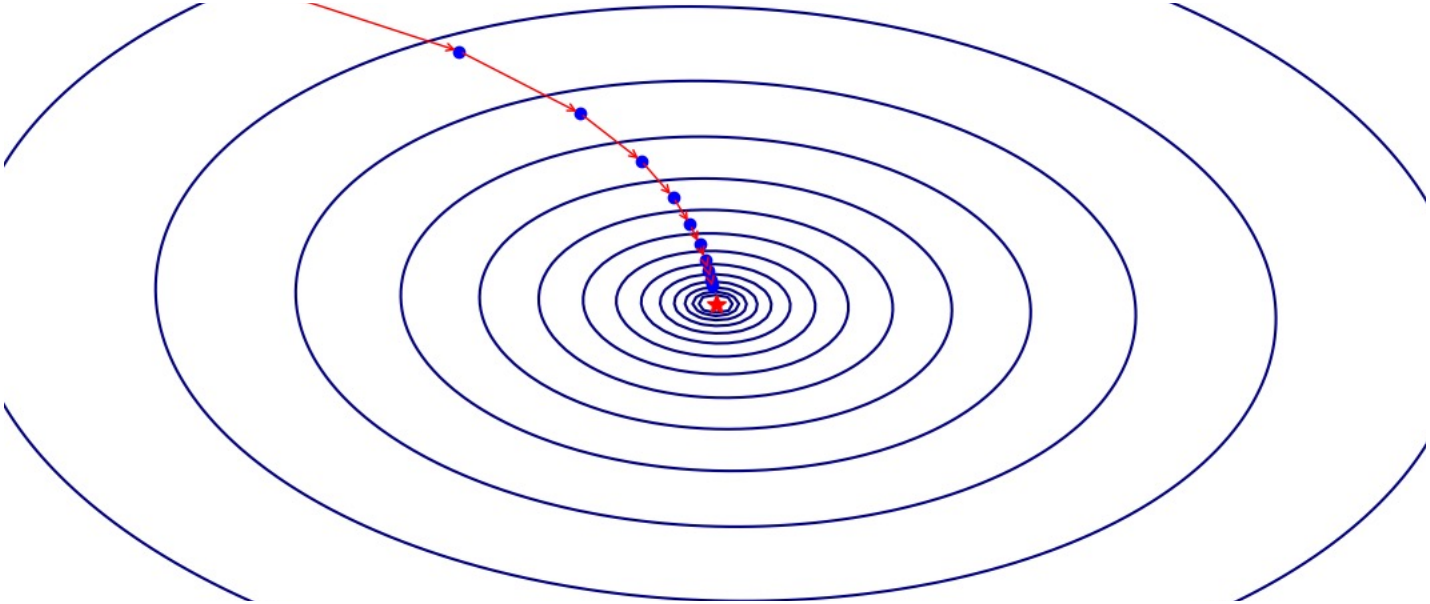


Advanced Topics in Deep Learning

Summer Semester 2024

4. Self-supervised Learning (Part II)

29.04.2024

Prof. Dr. Vasileios Belagiannis

Chair of Multimedia Communications and Signal Processing

Friedrich-Alexander-Universität
Technische Fakultät

Course Topics

Not for sharing (LMS, Friedrich-Alexander-Universität Erlangen-Nürnberg)

1. Interpretability.
2. Attention and Transformers.
3. **Self-supervised Learning.**
4. Similarity Learning.
5. Generative Models.
6. Model Compression.
7. Transfer learning, domain adaptation, few-shot learning.
8. Uncertainty Estimation.
9. Geometric Deep Learning.
10. Recap and Q&A.

- The exam will be written.
- We will have an exam preparation test.

Acknowledgements

- Special thanks Arij Bouazizi, Julia Hornauer, Julian Wiederer, Adrian Holzbock and Youssef Dawoud for contributing to the lecture preparation.

Recap

- Transfer learning.
- Self-supervised learning.
- Pretext tasks.
- Surrogate Classes.
- Patch Relationship.
- Jigsaw Puzzle.
- Rotation.
- Colorization.

Today's Agenda and Objectives

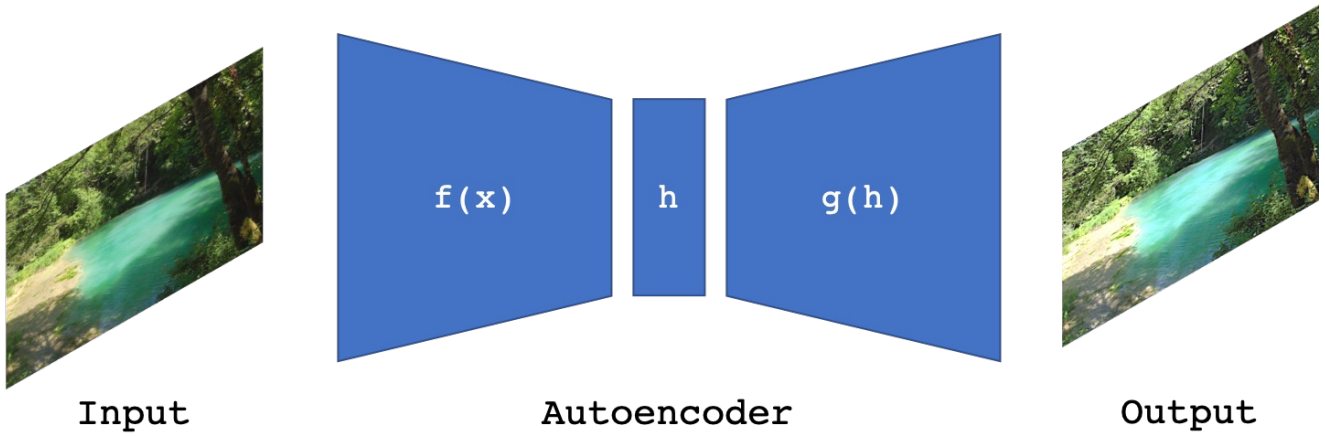
- Generative modelling.
- Contrastive learning loss functions.
- Self-supervised contrastive learning.
- Negative-free contrastive learning.
- Momentum contrastive learning.

Generative Modelling

- We can formulate pretext task in the contest of generative modelling.
- The pretext task can be to reconstruct the input while learning a meaningful representation.
- In the context of deep neural networks, an autoencoder performs this task.
- An autoencoder is a neural network that is trained to reconstruct its input. It is composed of the encoder, latent code and the decoder. The encoder $f(\cdot)$ transforms the input x to the latent code h, given by $\mathrm{h} = f(x)$. The decoder $\mathrm{g}(\cdot)$ reconstructs the input x from the latent code h, given by $\hat{x} = g(h)$.

Autoencoders

- An autoencoder has three major components.
- It can be trained as following:
  - Input: x, output: $\hat{x}$
  - Loss: $\mathcal{L} = \|x - \hat{x}\|^2$ where $\hat{x} = g(f(x))$.
- *What is the pretext task using an autoencoder?*



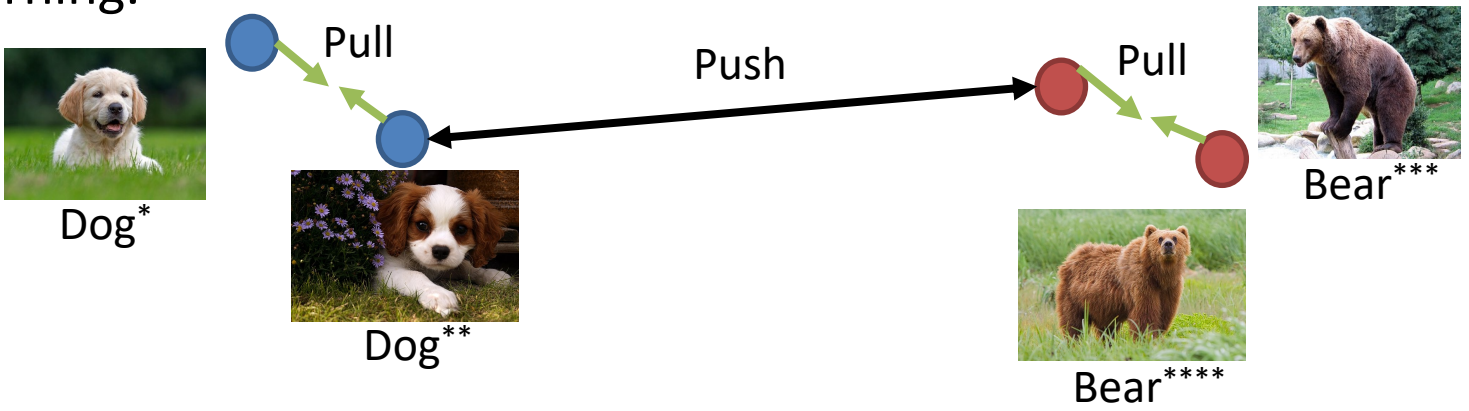
Denoising Autoencoders

- Input: It is partially corrupted by noise*. The noise comes from some prior distribution. This is a stochastic mapping of the clean signal x to a noisy signal $\hat{x}$, represented by:
  - $\hat{x} \sim q_D(\hat{x}|x)$.
  - The stochastic mapping $q_D(\cdot)$ can combine multiple types of prior distributions. In the simple case, it can set a number of elements to zero.
- Output: Given the noisy signal $\hat{x}$ as input, the output will be the clean signal x.
- The loss function for n training samples is defined as:
  - $\mathcal{L}_{DAE} = \frac{1}{n}\sum_{i=1}^{n}(x^i - g(f(\hat{x}^i)^2$.
- Training the denoising autoencoder is the pretext task for self-supervision.

*Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." Proceedings of the 25th international conference on Machine learning. 2008.

Contrastive Representation Learning

- The goal of contrastive learning is to learn an embedding space, i.e. a feature representation, in which similar pairs of samples, e.g. objects of the same class, are close together and dissimilar pairs are far apart.
- Contrastive learning can be applied to supervised or unsupervised learning.



- It is the de facto approach to perform self-supervision.
- It was developed in the context of generative modelling.

*By Hebrew Matio - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=95125027
**By leisergu - https://www.flickr.com/photos/leisergu/2873249326, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=34279814
***By Jean-noël Lafargue - own photo, w:fr:Parc animalier des Pyrénées, France, FAL, https://commons.wikimedia.org/w/index.php?curid=262707
****By Yathin S Krishnappa - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=23241619

Contrastive Loss

- Contrastive learning can be implemented with the contrastive loss.
- The idea is to learn a similarity metric for the face verification task*. This is one of the first work on the topic.
- Given the input x_i we aim to encode it to an embedding representation of d dimensions with the mapping function $f_\theta: \mathcal{X} \rightarrow \mathcal{R}^d$.
- The same class samples with x_i should lie close in the embedding space, while the samples of different class should lie as far as possible from x_i.
  - The constative loss implements these two conditions in a differentiable function that can be plugged into a deep neural network.

*Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.

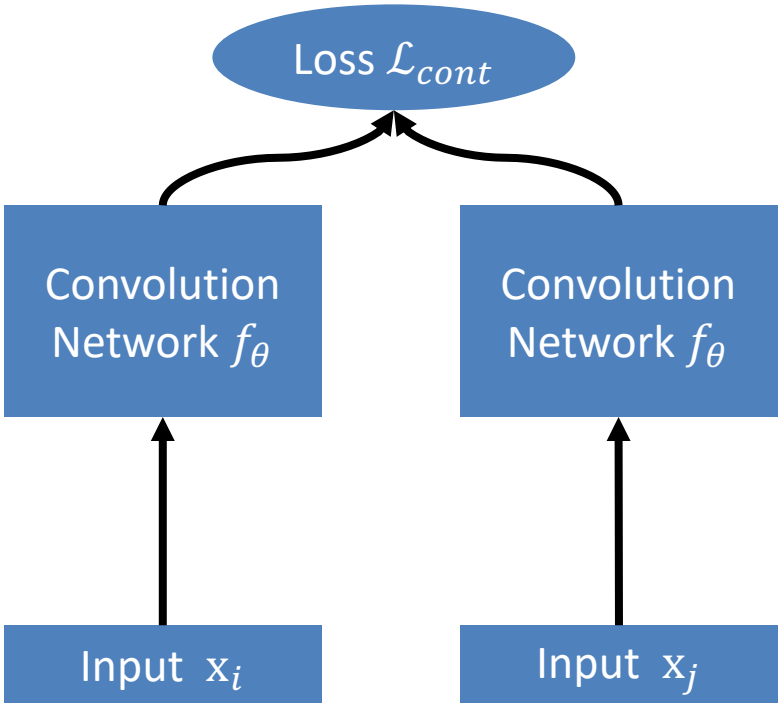
Contrastive Loss (Cont.)

- Given a pair of inputs $(\mathrm{x}_i, \mathrm{x}_j)$ the loss is defined as:
  - $\mathcal{L}_{cont}(\mathrm{x_i}, \mathrm{x_j}, \theta) = \mathbb{1}[y_i = y_j] \left\| f_\theta(\mathrm{x}_i) - f_\theta(\mathrm{x}_j) \right\|_2^2 + \mathbb{1}[y_i \neq y_j] \max(0, \epsilon - \left\| f_\theta(\mathrm{x}_i) - f_\theta(\mathrm{x}_j) \right\|_2^2)$.
  - where y_i and y_j are the corresponding labels of x_i and x_j, and ϵ is a hyperparameter defining the lower bound distance between samples of <u>different</u> classes.
  - ϵ is like a margin, similar to SVM.
  - $\mathrm{x}_i, \mathrm{x}_j$ is pair of images of the same class or different classes.
  - Both images go through the same neural network f_θ.
  - The reference sample x_i is compared only with another x_j. Consider that we could have created more pairs for the x_i.
- For the face recognition example, the positive pairs include images of the same person, while a negative pair has image of two different persons.

*Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.

Contrastive Loss (Cont.)

- The convolutional neural network used to implement contrastive loss is called a Siamese architecture.
- This is because the input pair passes through the same model with the same parameters.
- *How can the model be used for face recognition after contrast-loss training?*



*Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.

Noise Contrastive Estimation

- The noise contrastive estimation (NCE) loss was initially proposed to estimate parameters of unnormalized statistical models.
- It applies a nonlinear logistic regression to differentiate between observed samples (positive) and generated noisy samples (negative).
- Let x be the target (observed) sample $P(\mathrm{x}|C=1;\theta)=p_\theta(\mathrm{x})$ and $\tilde{\mathrm{x}}$ be a noise sample $P(\tilde{\mathrm{x}}|C=0;\theta)=q(\tilde{\mathrm{x}})$. The NCE loss is defined as :
  - $\mathcal{L}_{NCE}=\frac{-1}{N}\sum_{i=1}^{N}[\log(\sigma(l_\theta(\mathrm{x}_i))+\log(1-\sigma(l_\theta(\tilde{\mathrm{x}}_i)))]$.
  - where $l_\theta(u)=\log\frac{p_\theta(u)}{q(u)}$ and $\sigma(l)$ is a sigmoid function that converts logits into probabilities.
  - The above loss includes one positive and one negative sample.
- *How can we incorporate more samples in the loss?*

*Gutmann, M. & Hyvärinen, A.. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics.

InfoNCE

- The InfoNCE* loss is inspired by the NCE loss. It is similar to the N-pair loss in that it includes multiple negative samples in the calculation of the contrastive loss.
- It relies on the cross-entropy loss to discriminate the positive from a set of independent negative samples.
- State-of-the-art contrastive learning approaches rely InfoNCE-like loss functions.

*Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv preprint arXiv:1807.03748* (2018).

InfoNCE (Cont.)

- Given a context vector $\mathbf{c}$, a positive sample is drawn from the conditional distribution $p(\mathbf{x}|\mathbf{c})$.
- For simplicity, assume $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ contains all samples among which only one positive sample exists. The probability of sampling a positive sample correctly is given by:
  - $$p(C = \text{pos}|\mathbf{X}, \mathbf{c}) = \frac{p(\mathbf{x}_{\text{pos}}|\mathbf{c}) \prod_{i=1,\dots,N;i\neq\text{pos}} p(\mathbf{x}_i)}{\sum_{j=1}^{N} p(\mathbf{x}_j|\mathbf{c}) \prod_{i=1,\dots,N;i\neq j} p(\mathbf{x}_i)} = \frac{\frac{p(\mathbf{x}_{\text{pos}}|\mathbf{c})}{p(\mathbf{x}_{\text{pos}})}}{\sum_{j=1}^{N} \frac{p(\mathbf{x}_j|\mathbf{c})}{p(\mathbf{x}_j)}} = \frac{f(\mathbf{x}_{\text{pos}},\mathbf{c})}{\sum_{j=1}^{N} f(\mathbf{x}_j,\mathbf{c})}.$$
- InfoNCE loss uses the categorical cross-entropy loss to identify one positive sample among a set of negative samples. Hence, the loss is defined as:
  - $$\mathcal{L}_{InfoNCE} = -\mathbb{E}_{\mathbf{X}}\left[\log \frac{f(\mathbf{x},\mathbf{c})}{\sum_{\mathbf{x}'\in\mathbf{X}} f(\mathbf{x}',\mathbf{c})}\right].$$
- The loss optimizes the negative likelihood of classifying positive samples correctly.

*Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv preprint arXiv:1807.03748* (2018).

Soft-Nearest Neighbour Loss

- The soft-Nearest neighbour loss considers multiple positive samples.
- Assuming a batch of samples $\{(\mathrm{x}_i, y_i)\}_{i=1}^{B}$, where y_i is the label of x_i, the loss is defined as follows:
  - $\mathcal{L}_{snn} = \frac{-1}{B}\sum_{i=1}^{B}\log\frac{\sum_{i\neq j, y_i=y_j, j=1,\dots,B}\exp(-f(\mathrm{x}_i,\mathrm{x}_j)/\tau)}{\sum_{i\neq k, j=1,\dots,B}\exp(-f(\mathrm{x}_i,\mathrm{x}_k)/\tau)}.$
  - $f(.,.)$ is the function to measure the similarity between two inputs.
  - τ is a temperature (scaling) parameter that tunes how well features are concentrated in features space.
  - For example, a low temperature value allows the low distance to dominate the loss, i.e. the loss focuses more on the low distance compared to widely separated representations.
- *Can we work without knowing the input labels?*

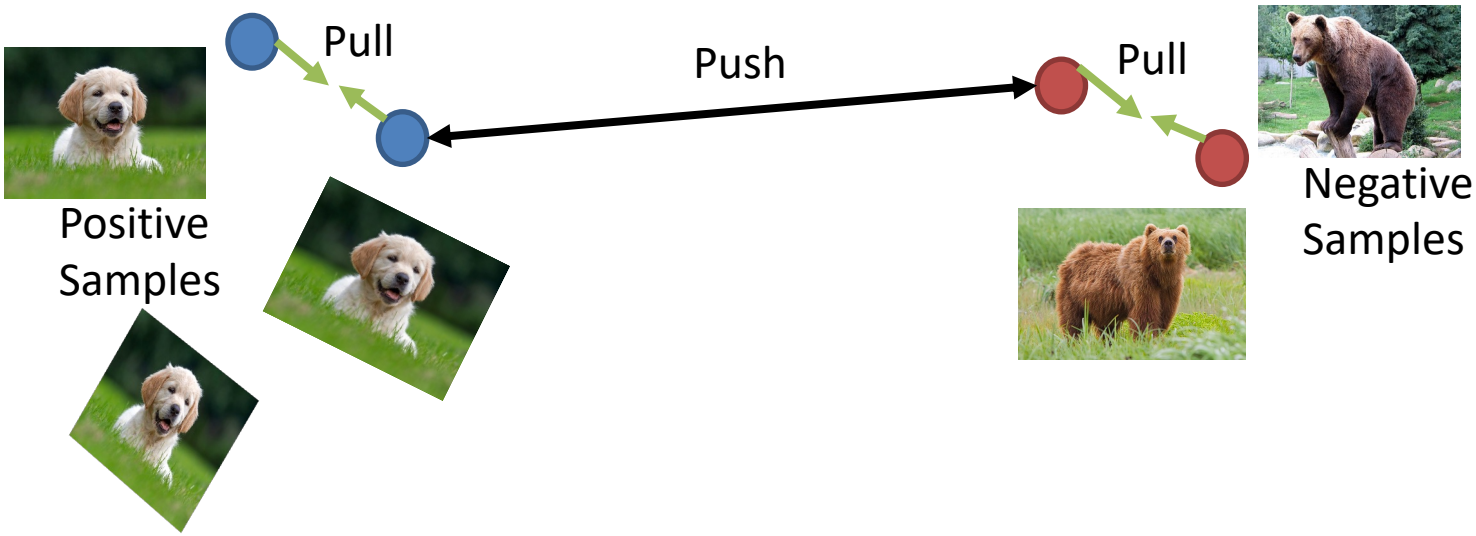
*Frosst, Nicholas, Nicolas Papernot, and Geoffrey Hinton. "Analyzing and improving representations with the soft nearest neighbor loss." International conference on machine learning. PMLR, 2019.

From Supervised to Self-Supervised Learning

- Contrastive learning requires positive and negative labels, e.g. Contrastive loss and NCE loss.
- The label does not necessarily need to be an object category.
- For example, each sample of the dataset could be given a unique label, and we could use data augmentation to create multiple instances of it. This could be the positive label (class). Furthermore, the rest of the dataset can form the negative labels (classes).
- For contrastive learning, the positive and negative samples can be defined in this way. A positive sample would be any data instance and its augmentations, while a negative sample would be all other data instances and their augmentations.
- *What are the challenges in defining positive and negative samples?*

Positive and Negative Samples

- Each sample forms a unique object class. The positive samples are generated from the data augmentation and the negatives from the remaining samples.



Self-Supervised Learning Challenges

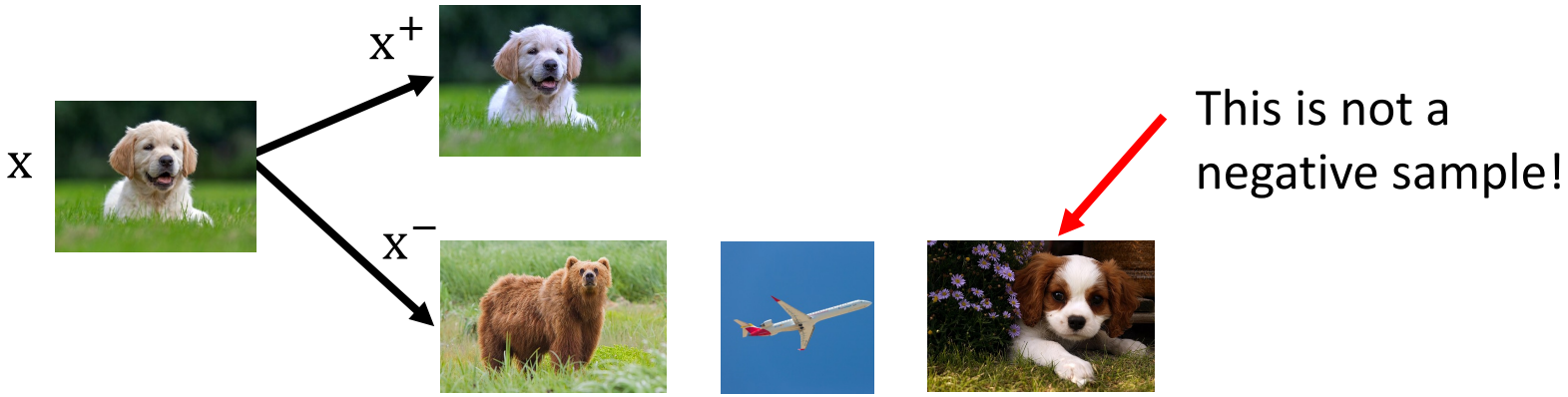
- Appropriate data augmentation is needed so that the semantics of the positive samples are not heavily perturbed in order to learn useful representations of the data.
- Common image-based augmentations include:
  - Blurring, colour jittering, horizontal flipping.
  - This kind of augmentations are cheap to apply.
  - In addition, augmentation can include rotation, translation, crop, cutmix and/or mixup.
- Self-Supervised is normally based on backpropagation and gradient descent standard training. However, the number of positive and negative samples within the processed mini-batch affects the gradient updates.

Self-Supervised Learning Challenges (Cont.)

- The minimum mini-batch size has an impact on many contrastive learning methods, in particular those that rely on negative samples in the batch.
- A large mini-batch size includes a large number of negative samples that are challenging enough for the model to learn meaningful representations and distinguish between different examples.
- *What is the challenge of a large mini-batch?*

Self-Supervised Learning Challenges (Cont.)

- Hard negative mining is the third main challenge in self-supervised learning.
  - Meaningful (hard) negatives are necessary to learn a robust feature representation. Increasing the mini-batch size can help to obtain hard negatives.
  - However, sampling without labels carries the risk of false negative samples, e.g. positive and negative class includes a dog. This is known as sampling bias*.



*Chuang, Ching-Yao, et al. "Debiased contrastive learning." *Advances in neural information processing systems* 33 (2020).

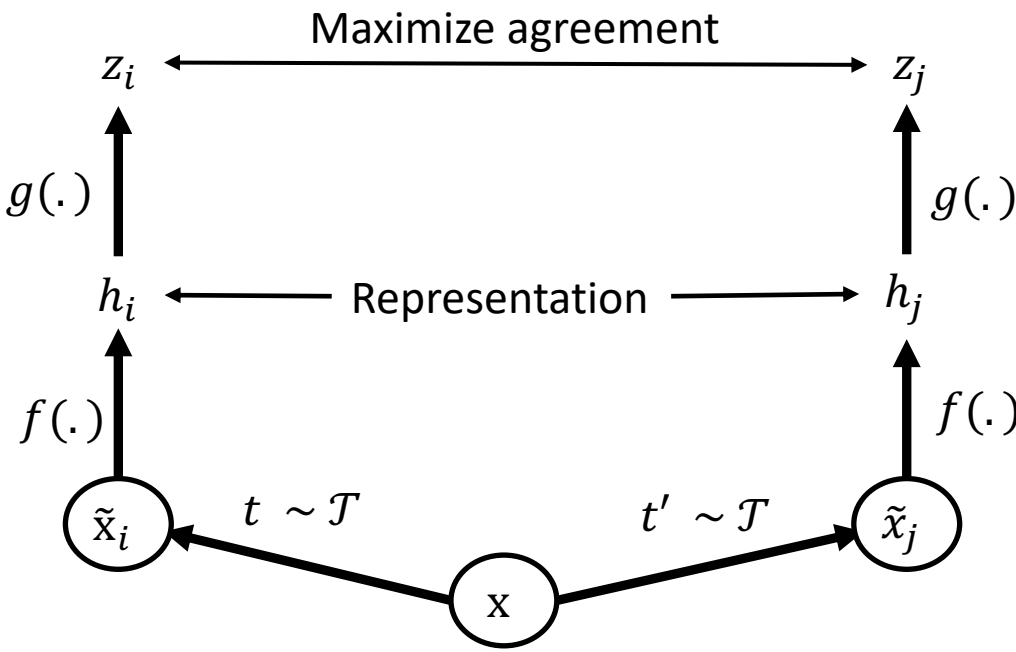
Simple Contrastive Learning

- The simple framework for contrastive learning (SimCLR*) of visual representations is a popular approach for contrastive learning.
- The SimCLR* algorithm learns a visual representation by maximizing the agreement between differently augmented views of the same sample via the contrastive loss in the latent space.
- The framework is based on:
  - Data augmentation.
  - Neural network base encoder $f(.)$.
  - Neural network projection head $g(.)$.
  - Contrastive loss function.

*Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

SimCLR Algorithm

- A mini-batch of size N is randomly sampled.
- Each image x in the mini-batch is transformed twice:
  - $\tilde{\mathrm{x}}_i = t(\mathrm{x})$,
  - $\tilde{\mathrm{x}}_j = t'(\mathrm{x})$,
  - With $t, t' \sim \mathcal{T}$.
  - where t and t' are sampled from the same augmentation policy $\mathcal{T}$. After augmentation, the effective mini-batch size is doubled i.e. $2N$.
- Given one positive pair and $2(N-1)$ negative pairs, the representations are extracted using backbone network $f(.)$ as:
  - $h_i = f(\tilde{\mathrm{x}}_i)$, $h_j = f(\tilde{\mathrm{x}}_j)$
  - where h_i , h_j are raw representations of dimension D.



SimCLR Algorithm (Cont.)

- The contrastive loss for each positive pair is computed as:
  - $\mathcal{L}_{SimCLR}^{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1[k \neq i] \exp(\text{sim}(z_i, z_k)/\tau)}$.
  - $1[k \neq i]$ is an indicator function i.e. 1 if $k \neq i$ and 0 otherwise.
  - $\text{sim}(.,.)$ is a cosine similarity score that operates on the output of $g(.)$.
- Note that $\mathcal{L}_{SimCLR}^{i,j}$ is simply InfoNCE loss where we have multiple negative samples and one pair of positive sample.
- We use the backbone from SimCLR pre-training in the downstream task and throw away $g(.)$.
- In the paper, it was empirically shown that adding a projection head $g(.)$ to transform raw representations into a lower dimensional space d, where $d < D$, improves contrastive learning performance.
  - $z_i = g(h_i)\,, z_j = g(h_j)$.
  - z is the projected feature onto the lower dimensional head.

SimCLR Results

- The linear evaluation corresponds to fixing pretrained backbone parameters while training task head only.
- Fine-tuned denotes training the entire network.
- Supervised means using ImageNet pretrained weights i.e., supervised (transfer) pre-training using labels.
- Random Init means using randomly initialized parameters i.e., no pretraining.

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
SimCLR (ours)	**76.9**	**95.3**	80.2	48.4	**65.9**	60.0	61.2	**84.2**	**78.9**	89.2	**93.9**	**95.0**
Supervised	75.2	**95.7**	**81.2**	**56.4**	64.9	**68.8**	**63.8**	83.8	**78.7**	**92.3**	**94.1**	94.2
Fine-tuned:												
SimCLR (ours)	**89.4**	**98.6**	**89.0**	**78.2**	**68.1**	**92.1**	**87.0**	**86.6**	**77.8**	92.1	**94.1**	97.6
Supervised	88.7	98.3	**88.7**	**77.8**	67.0	91.4	**88.0**	86.5	**78.8**	**93.2**	**94.2**	**98.0**
Random init	88.3	96.0	81.9	**77.0**	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

SimCLR Limitations

- Large mini-batch sizes consume memory. This in turn makes each sample in the mini-batch expensive to process.
- Performance is vulnerable to augmentation policy.
- The contrastive learning loss requires negative samples, which can introduce sampling error, but there are other algorithms that use a negative-free contrastive loss.

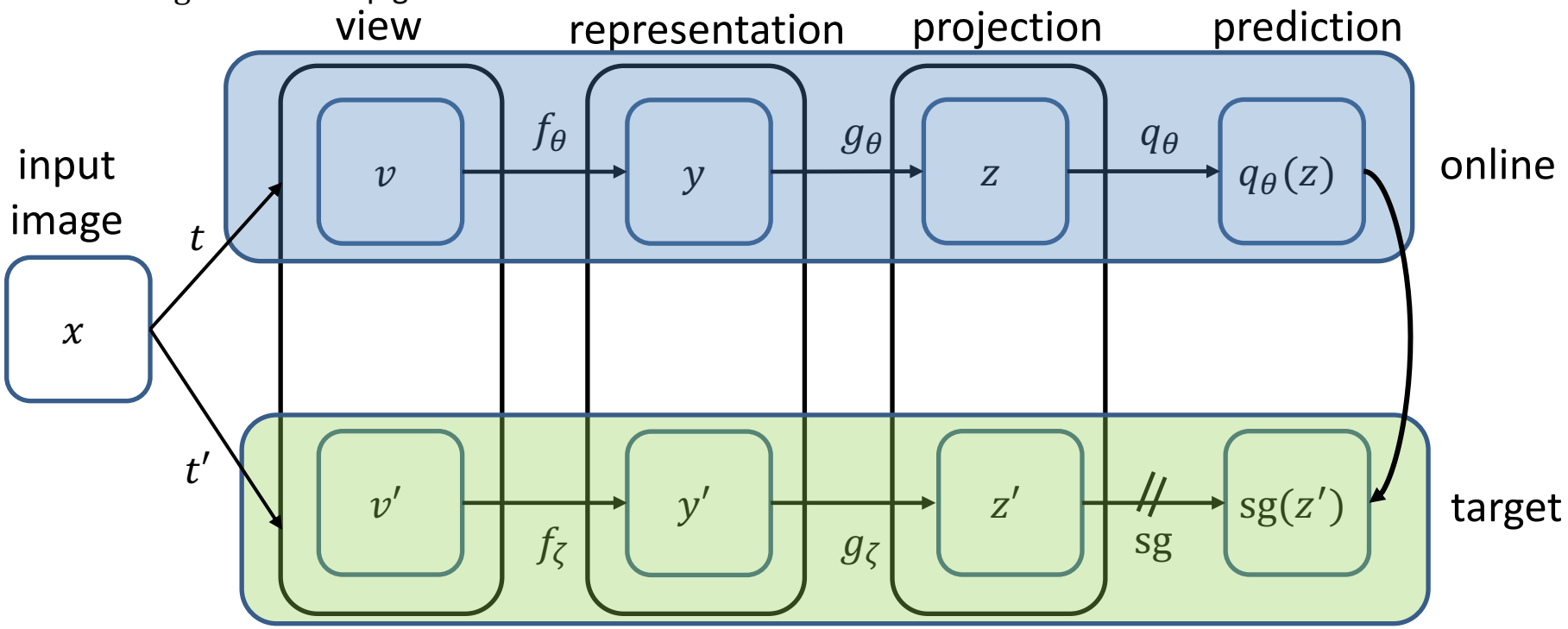
Negative-Free Contrastive Learning

- Bootstrap Your Own Latent (BYOL) is a contrastive learning method that does not require negative samples.
- It is based on two neural networks, the online network and the target network. The online network learns to predict features extracted by the target network from an input image under a different augmentation (view).
- Both networks share the same architecture.
- The learned representation can be then fine-tune to a downstream task, similar to the other self-supervised approaches.

*Grill, Jean-Bastien, et al. "Bootstrap your own latent-a new approach to self-supervised learning." Advances in neural information processing systems 33 (2020): 21271-21284.

BYOL

- The online network, parametrized by θ, consists of the encoder f_θ, projection head g_θ, and the predictor q_θ. It is updated with gradient descent.
- The target network has different parameters ζ and it is updated by the exponential moving average i.e. $\zeta \leftarrow \tau\zeta + (1-\tau)\theta$.
- sg refers to stop gradient.



*Grill, Jean-Bastien, et al. "Bootstrap your own latent-a new approach to self-supervised learning." Advances in neural information processing systems 33 (2020): 21271-21284.

BYOL Algorithm

- Create two augmented views $v = t(x), v' = t'(x)$, with augmentation sampled from $t \sim \mathcal{T}, t' \sim \mathcal{T}$.
- Extract raw representation $y = f_\theta(v), y' = f_\zeta(v')$.
- Project into latent variables $z = g_\theta(v), z' = g_\zeta(v')$.
- The online network outputs prediction $q_\theta(z)$. Note that both $q_\theta(z)$ and z' are L2-normalized $\bar{q}_\theta(z) = \frac{q_\theta(z)}{|q_\theta(z)|}$, $\bar{z}' = \frac{z'}{|z'|}$.
- Calculate the BYOL-loss (mean-squared error) $\mathcal{L}_{BYOL} = \left\| \bar{q}_\theta(z) - \bar{z}' \right\|_2^2$
- The online network is updated using an optimizer e.g. SGD while target network is updated using exponential moving average of θ.

*Grill, Jean-Bastien, et al. "Bootstrap your own latent-a new approach to self-supervised learning." Advances in neural information processing systems 33 (2020): 21271-21284.

BYOL Results

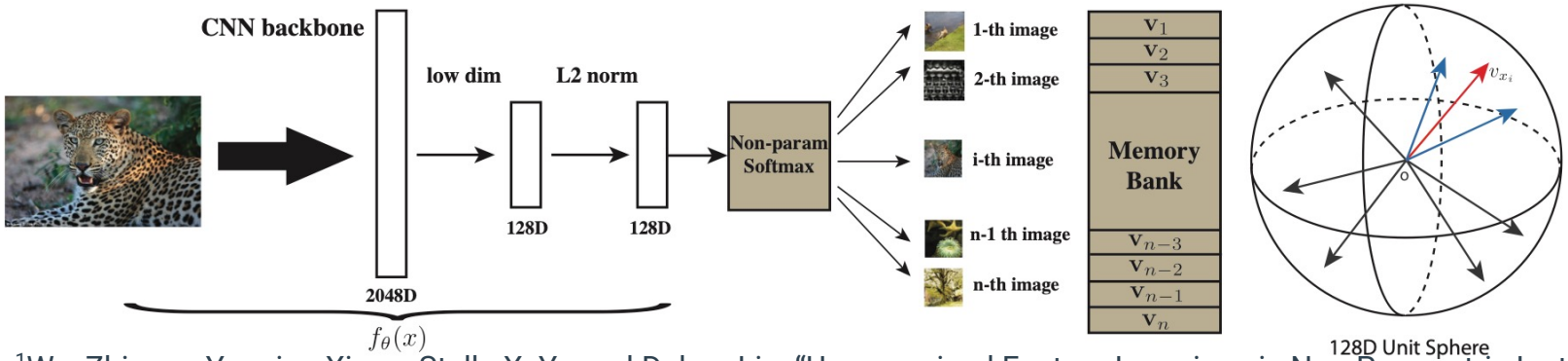
Not for sharing (LMS, Friedrich-Alexander-Universität Erlangen-Nürnberg)

- The approach works surprisingly well for not using negative samples.
- A follow-up study showed that BYOL can work as random sampling if one remove the batch normalisation from the neural networks. Using negative samples remains important to avoid mode collapse. More information at the following blog: https://generallyintelligent.com/blog/2020-08-24-understanding-self-supervised-contrastive-learning/

Method	Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
BYOL (ours)	**75.3**	91.3	**78.4**	**57.2**	**62.2**	**67.8**	60.6	82.5	75.5	90.4	94.2	**96.1**
SimCLR (repro)	72.8	90.5	74.4	42.4	60.6	49.3	49.8	81.4	**75.7**	84.6	89.3	92.6
SimCLR [8]	68.4	90.6	71.6	37.4	58.8	50.3	50.3	80.5	74.5	83.6	90.3	91.2
Supervised-IN [8]	72.3	**93.6**	78.3	53.7	61.9	66.7	**61.0**	**82.8**	74.9	**91.5**	**94.5**	94.7
Fine-tuned:												
BYOL (ours)	**88.5**	**97.8**	86.1	**76.3**	63.7	91.6	**88.1**	**85.4**	**76.2**	91.7	**93.8**	97.0
SimCLR (repro)	87.5	97.4	85.3	75.0	63.9	91.4	87.6	84.5	75.4	89.4	91.7	96.6
SimCLR [8]	88.2	97.7	85.9	75.9	63.5	91.3	88.1	84.1	73.2	89.2	92.1	97.0
Supervised-IN [8]	88.3	97.5	**86.4**	75.8	**64.3**	**92.1**	86.0	85.0	74.6	**92.1**	93.3	**97.6**
Random init [8]	86.9	95.9	80.2	76.1	53.6	91.4	85.9	67.3	64.8	81.5	72.6	92.0

Instance contrastive learning

- Each instance of the dataset is considered as distinct class of its own.
- The size of the dataset will the be the number classes.
- Given the softmax output and the cross-entropy loss,, the task is to classify each class.
  - In practice, it is not possible to have a softmax with hundreds of thousands or even million(s) of outputs.
  - Instead the NCE loss on the logits is used to approximate the softmax output.



[1]Wu, Zhirong, Yuanjun Xiong, Stella X. Yu and Dahua Lin. "Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination." *CVPR* (2018).

Instance contrastive learning (Cont.)

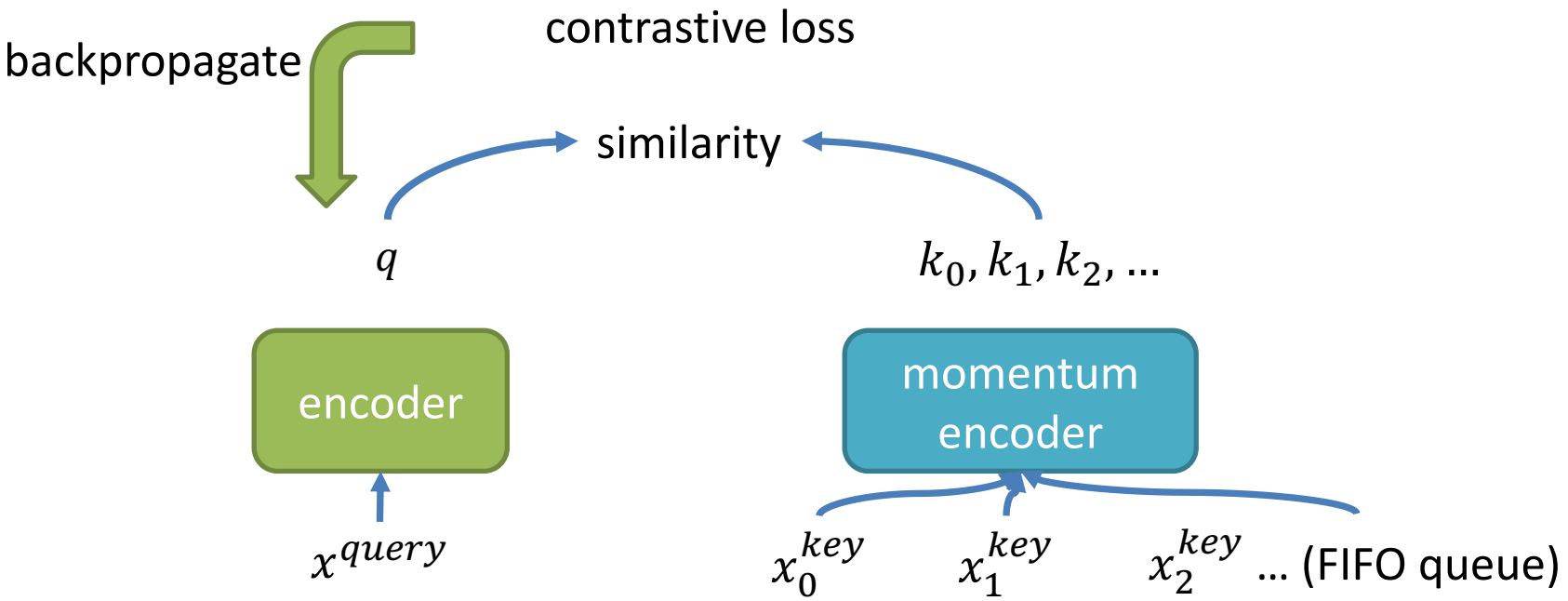
- Let $\mathrm{v} = f_\theta(\mathrm{x})$ be the embedding function to learn and the vector v is normalized. A non-parametric classifier is employed to predict the probability of sample v belonging to class i as:
  - $P(C = i|\mathrm{v}) = \frac{\exp(\mathrm{v}_i^T \mathrm{v}/\tau)}{\sum_{j=1}^{n} \exp(\mathrm{v}_j^T \mathrm{v}/\tau)}$.
  - where τ is a temperature parameter.
- Note that the denominator in $P(C = i|\mathrm{v})$ requires to access to representations of all samples, which is an expensive computation.
- Instead, a memory bank is used to store feature representations from past iterations in a database instead of computing it every time.
- Let $\mathrm{V} = \{\mathrm{v}_\mathrm{i}\}$ be the memory bank and $\mathrm{f}_i = f_\theta(\mathrm{x}_i)$ be the features generated by forwarding the input image through network. We can rely on the features stored in V instead of performing forward pass every iteration to get f_i.
- Finally, we can approximate the denominator in $P(C = i|\mathrm{v})$ with Monte Carlo approximation using a random subset of M indices $\{j_k\}_{k=1}^{M}$ given as:
  - $P(C = i|\mathrm{v}) = \frac{\exp(\mathrm{v}^T \mathrm{f}_i/\tau)}{\sum_{j=1}^{n} \exp(\mathrm{v}_j^T \mathrm{f}_i/\tau)} = \frac{\exp(\mathrm{v}^T \mathrm{f}_i/\tau)}{\frac{N}{M}\sum_{k=1}^{M} \exp(\mathrm{v}_{jk}^T \mathrm{f}_i/\tau)}$.

Instance contrastive learning (Cont.)

- Since there is only one instance per class, the training can be unstable and may severely fluctuate.
- To regularise the training process, an additional term is introduced for positive samples in the loss function. The final objective is:
  - $\mathcal{L}_{instance} = -\mathbb{E}_{P_d}\left[\log h\left(i, \mathrm{v}_i^{t-1}\right) - \lambda\left\|\mathrm{v}_i^t - \mathrm{v}_i^{t-1}\right\|_2^2\right] - M\mathbb{E}_{P_n}\left[\log\left(1 - h\left(i, \mathrm{v}_i^{t-1}\right)\right)\right]$
  - with $h\left(i, \mathrm{v}_i^{t-1}\right) = \frac{P(C=i|\mathrm{v})}{P(C=i|\mathrm{v}) + MP_n(C=i)}$.
  - $\{\mathrm{v}_i^{t-1}\}$ are the representations stored from previous iterations,
  - The noise distribution is uniform $P_n = \frac{1}{N}$.
  - The difference $\left\|\mathrm{v}_i^t - \mathrm{v}_i^{t-1}\right\|$ diminishes gradually as learning the embedding converges.

Momentum Contrastive Learning

- The Momentum Contrast (MoCo*) approach relies also on the memory bank to store feature representations.
- The memory bank is in the form of a dynamic dictionary lookup, structured as a large FIFO queue of representations of all samples.
- It achieves faster access than the instance contrastive learning approach.



*He, Kaiming, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.

Momentum Contrastive Learning (Cont.)

- Given a query sample x_q , we obtain the query representation by forwarding the sample through the query encoder $\mathrm{q} = f_q(\mathrm{x}_q)$.
- Additionally, a list of key representations $\{\mathrm{k}_0, \mathrm{k}_1, \dots\}$ extracted by the momentum encoder are stored in the dictionary $\mathrm{k}_i = f_k(\mathrm{x}_i^k)$.
- The momentum encoder parametrized by θ_k is progressively updated by an exponential moving average (EMA) of the encoder parametrized by θ_q, both networks have the same architecture. The update step is given by:
  - $\theta_k \leftarrow m\theta_k + (1-m)\theta_q$.
- Assume there is one positive key k^+ in the dictionary that matches q. k^+ is created through a noisy (augmented) replicate of x_q. The loss of MoCo is defined as:
  - $\mathcal{L}_{MoCo} = -\log \frac{\exp(\mathrm{q}.\mathrm{k}^+/\tau)}{\sum_{i=1}^{N} \exp(\mathrm{q}.\mathrm{k}_i/\tau)}$

Momentum Contrastive Learning (Cont.)

- MoCo has an advantage over SimCLR in that it decouples the batch size from the number of negative samples by using a dictionary (memory bank). Therefore, its performance is not dependent on large batch sizes.
- The memory bank of the MoCo differs from the instance discrimination, the FIFO structure of the queue allows to use the representations of the immediately preceding mini-batch.
- MoCo-v2* extends MoCo by incorporating designs from SimCLR, namely adding a projection head after the encoder and applying stronger data augmentation. This has been shown to improve the performance of MoCo.

Study Material

Not for sharing (LMS, Friedrich-Alexander-Universität Erlangen-Nürnberg)

- *Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.*
- *Gutmann, M. & Hyvärinen, A.. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics.*
- *Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.*
- *He, Kaiming, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.*
- *Blog: https://lilianweng.github.io/posts/2019-11-10-self-supervised/*
- *Blog: https://lilianweng.github.io/posts/2021-05-31-contrastive/*
- *Blog: https://anilkeshwani.github.io/CPC/*

Next Lecture

Metric Learning