Transformer

Adrien Guille (ERIC @ Univ Lyon 2)

Rencontre SciDoLySE - 10 mai 2021

Apprentissage statistique à partir du texte

- **Approche « classique »**
  - Un texte : sac de mots
    - Ignore la sémantique des mots et la syntaxe des textes
  - Classifieurs usuels adaptés
    - Classifieur bayésien naïf multinomial
    - Régression logistique avec pénalités L1 et L2 (*i.e.* ElasticNet)
    - *Etc*.

Apprentissage statistique à partir du texte

- **Popularisation de l'apprentissage profond (milieu années 2010)**
    - Un texte : séquence de vecteurs-mots
    - Vecteurs-mots pré-entraînés (ou pas), spécialisés sur la tâche (ou pas)
- Réseaux de neurones profonds ad-hoc
    - Réseau de neurones convolutifs (*i.e.* CNN)
    - Réseau de neurones récurrent (*i.e.* RNN)

Apprentissage statistique à partir du texte

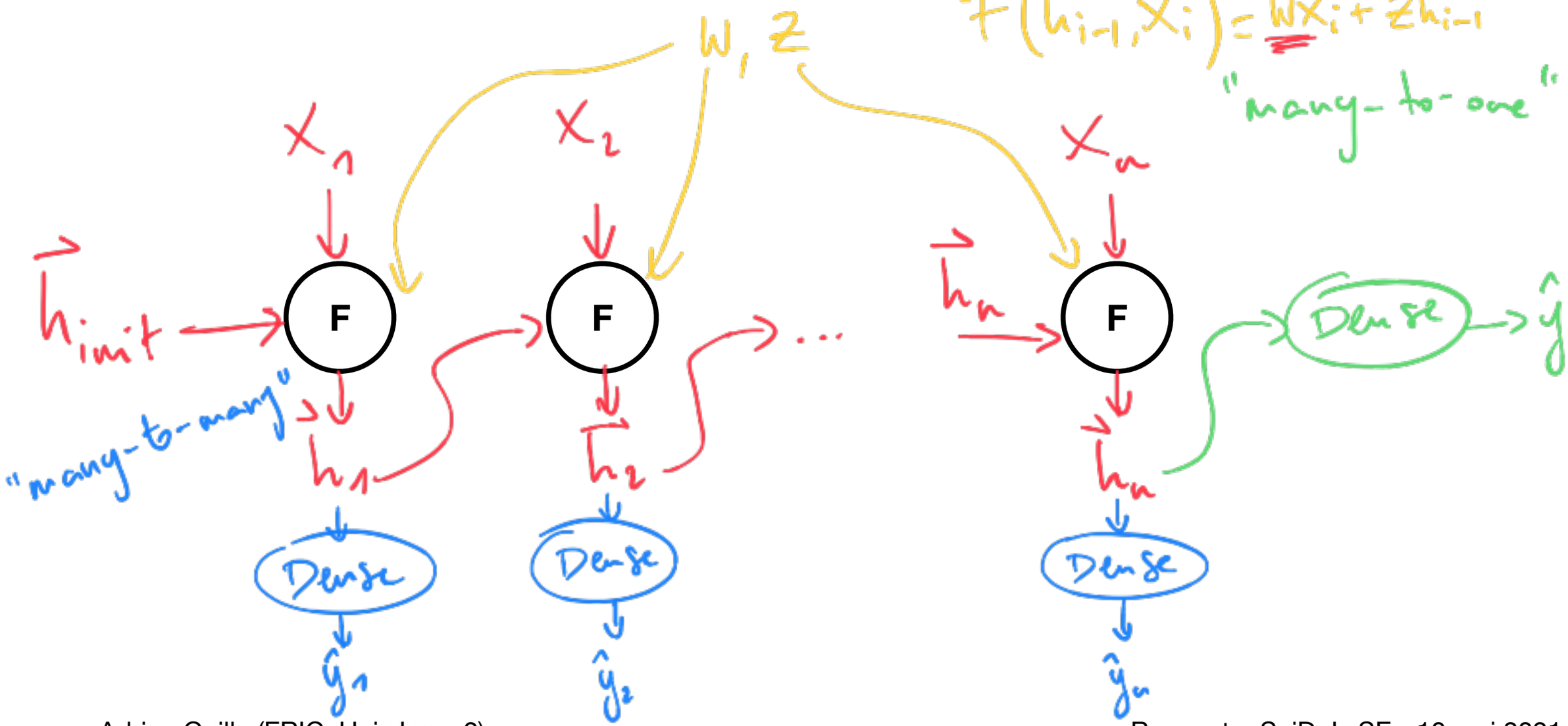
- **Avènement des modèles de langue pré-entraînés (début années 2020)**
    - Un texte : séquence de vecteurs-mots
    - Réseau de neurones « end-to-end »
        - Modèle de langue basé sur le Transformer (*e.g.* BERT, GTP-x)
        - Pré-entraîné une fois pour toute sur des tâches bas niveau
        - Spécialisé sur la tâche à résoudre

Des réseaux récurrents, à l'architecture encodeur-décodeur, au mécanisme d'attention

Réseaux de neurones récurrents

- **Architecture de base**

  - Entrée : $X \in \mathbb{R}^{n \times d}$

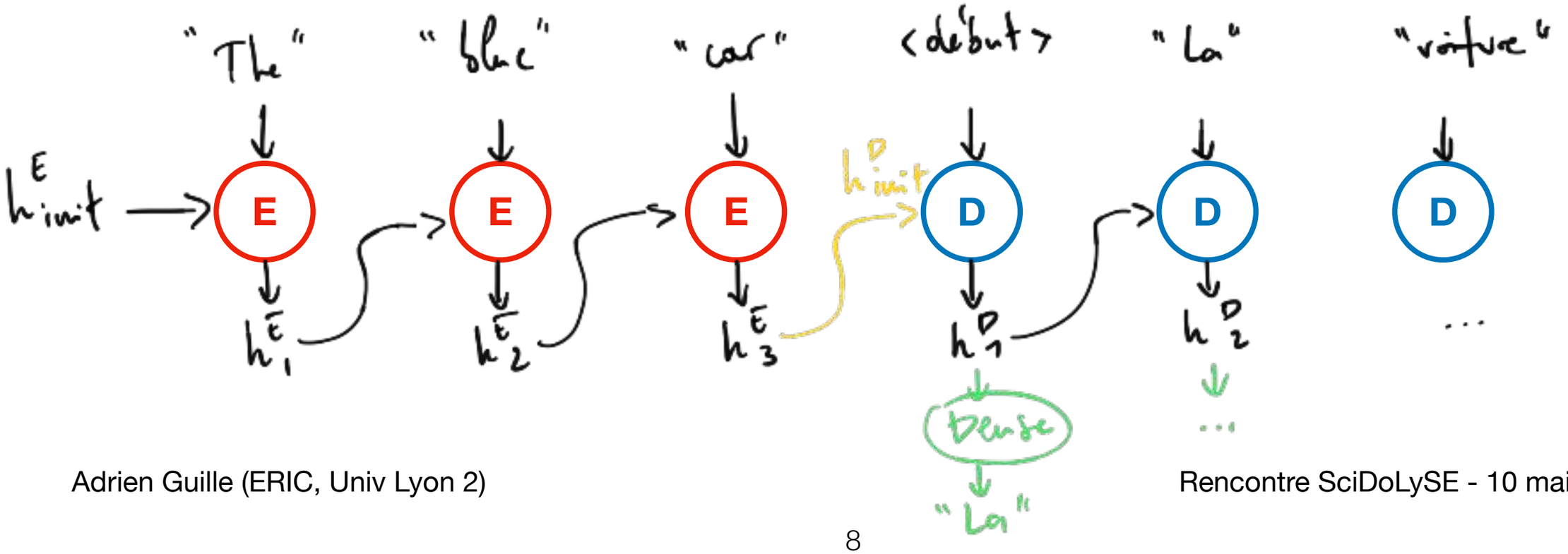


Réseaux de neurones récurrents

- **Inconvénients**
  - Propagation séquentielle : rend la parallélisation compliquée
  - Evanescence ou explosion du gradient : rend l'entraînement difficile voire impossible
- **Architecture Long/Short-Term Memory (LSTM)**
  - *Long Short-term Memory* (Hochreiter, Schmidhuber @ NeCo 9(8), 1997)
- **Astuces**
  - *E.g.* Rétropropagation tronquée, seuillage du gradient
  - *On the difficulty of training recurrent neural networks* (Pascanu, Mikolov, Bengio @ ICML 2013)

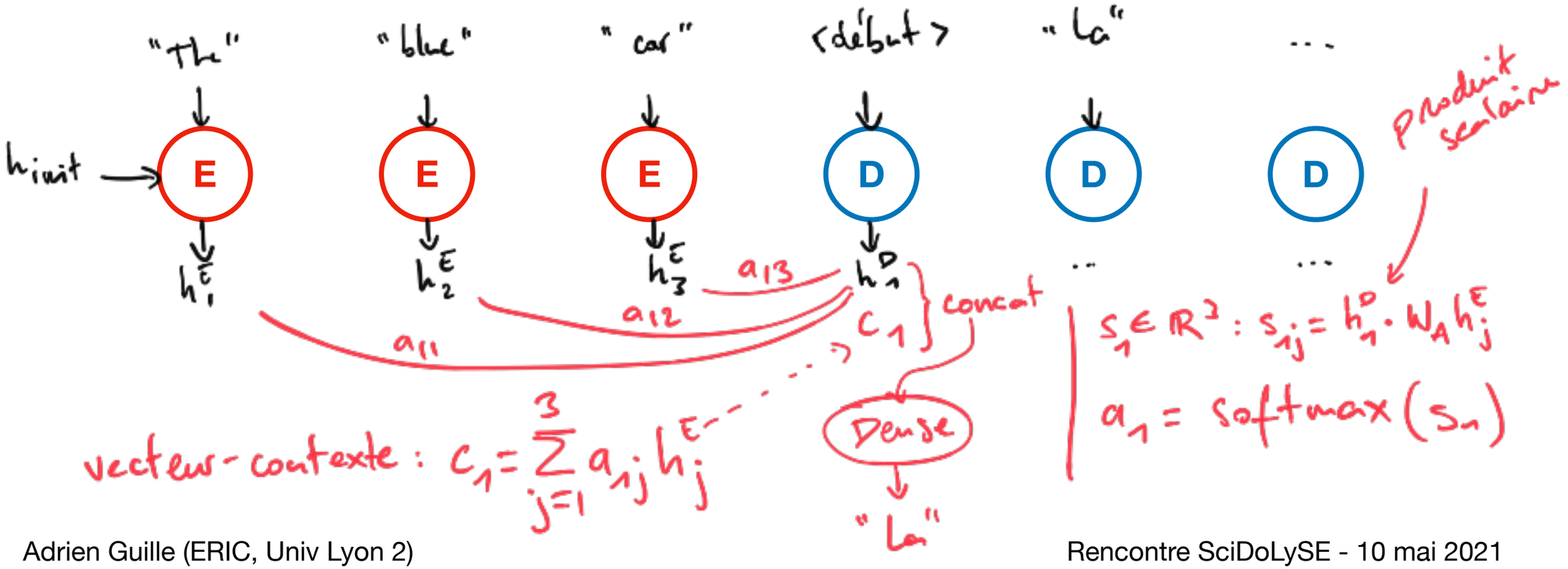
Encodeur-Décodeur

- **Architecture**
  - *Sequence to sequence learning with neural networks* (Sutskever, Vinyal, Le @ NeurIPS 2014)
  - Entrée : $X \in \mathbb{R}^{n \times d}$
  - Sortie : $Y = \langle y_1, y_2, \ldots, y_m \rangle$



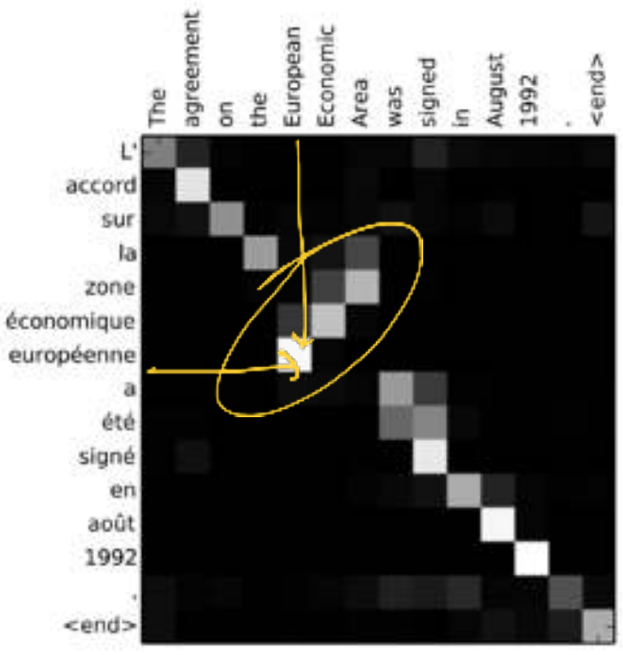
Encodeur-Décodeur + Attention

- **Lien direct entre états cachés de l'encodeur et du décodeur**
  - *Neural machine translation by jointly learning to align and translate* (Bahdanau, Cho, Bengio @ ICLR 2015)
  - *Effective approaches to attention-based neural machine translation* (Luong, Pham, Manning @ EMNLP 2015)



Encodeur-Décodeur + Attention

- **Avantages**
  - Gain qualitatif
  - Pallie le problème d'oubli progressif



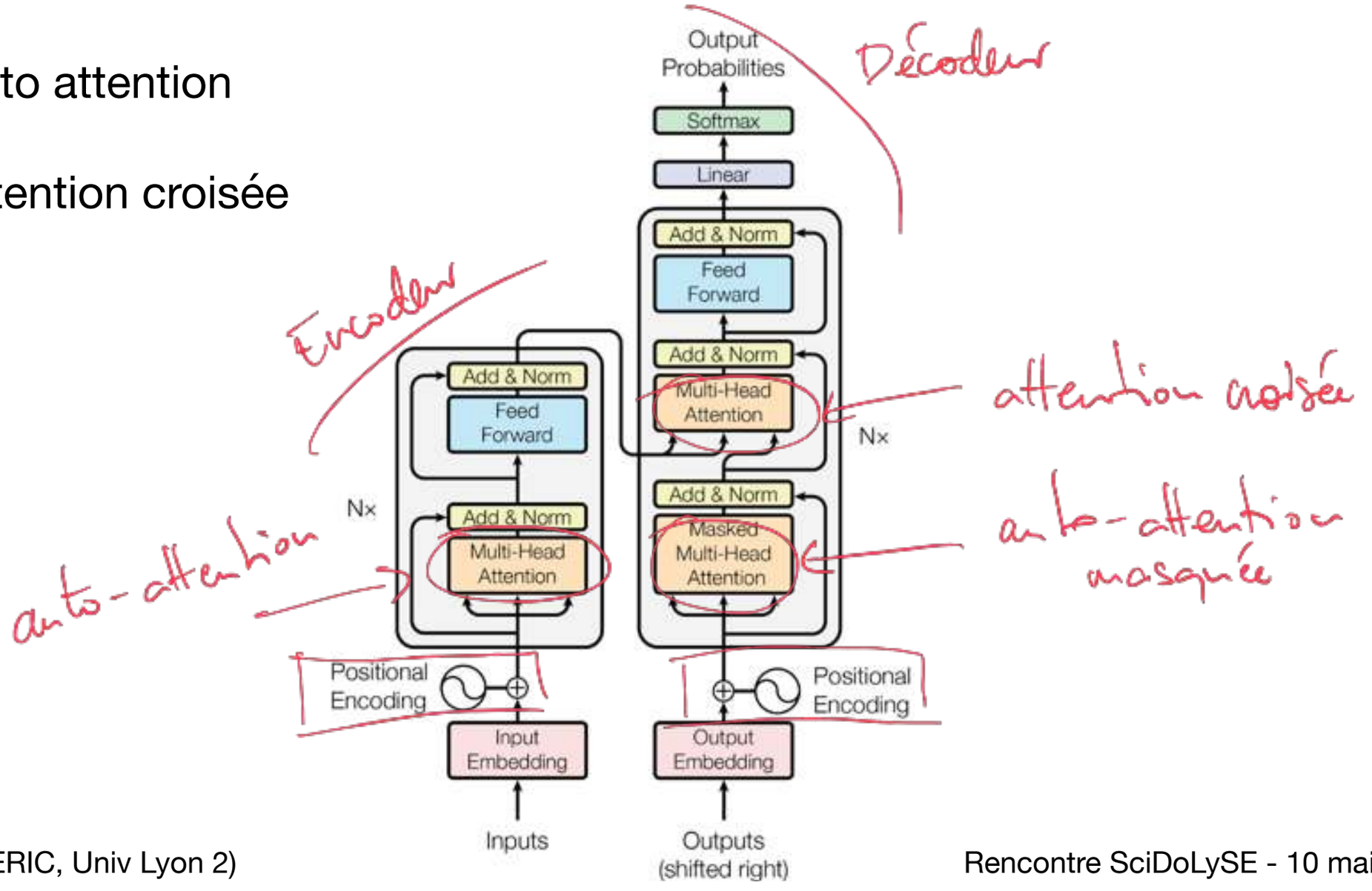
src	Wegen der von Berlin und der Europäischen Zentralbank verhängten strengen Sparpolitik in Verbindung mit der Zwangsjacke , in die die jeweilige nationale Wirtschaft durch das Festhalten an der gemeinsamen Währung genötigt wird , sind viele Menschen der Ansicht , das Projekt Europa sei zu weit gegangen
ref	The *austerity imposed by Berlin and the European Central Bank , coupled with the straitjacket* imposed on national economies through adherence to the common currency , has led many people to think Project Europe has gone too far .
best	Because of the strict *austerity measures imposed by Berlin and the European Central Bank in connection with the straitjacket* in which the respective national economy is forced to adhere to the common currency , many people believe that the European project has gone too far .

- **Inconvénient**
  - Parallélisation limitée du fait des RNN

Architecture du réseau Transformer

Transformer

- **Architecture générale**
  - *Attention is all you need* (Vaswani *et al.* @ NeurIPS 2017)
  - Auto attention
  - Attention croisée



Transformer : Encodeur

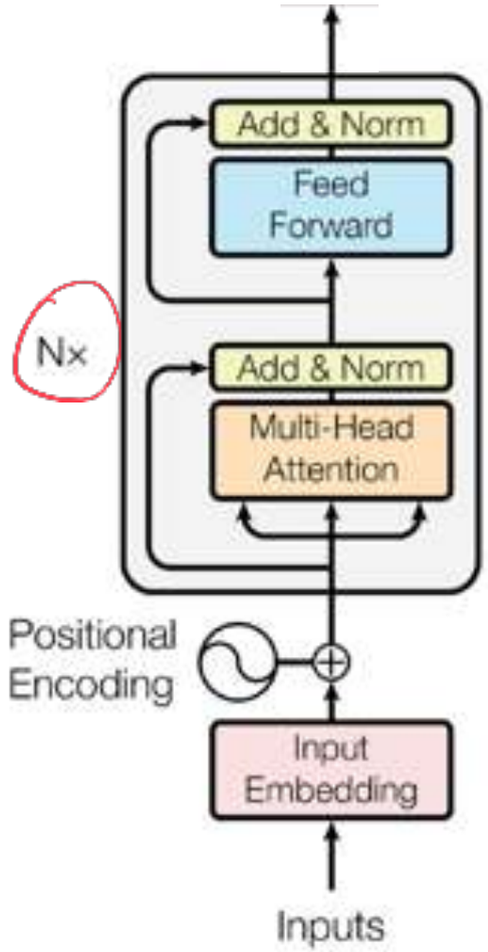
- **Encodage des positions en entrée**

$$X' = X \in \mathbb{R}^{n \times d} + P \in \mathbb{R}^{n \times d}$$

  - Somme des vecteur-mots et de vecteurs-positions

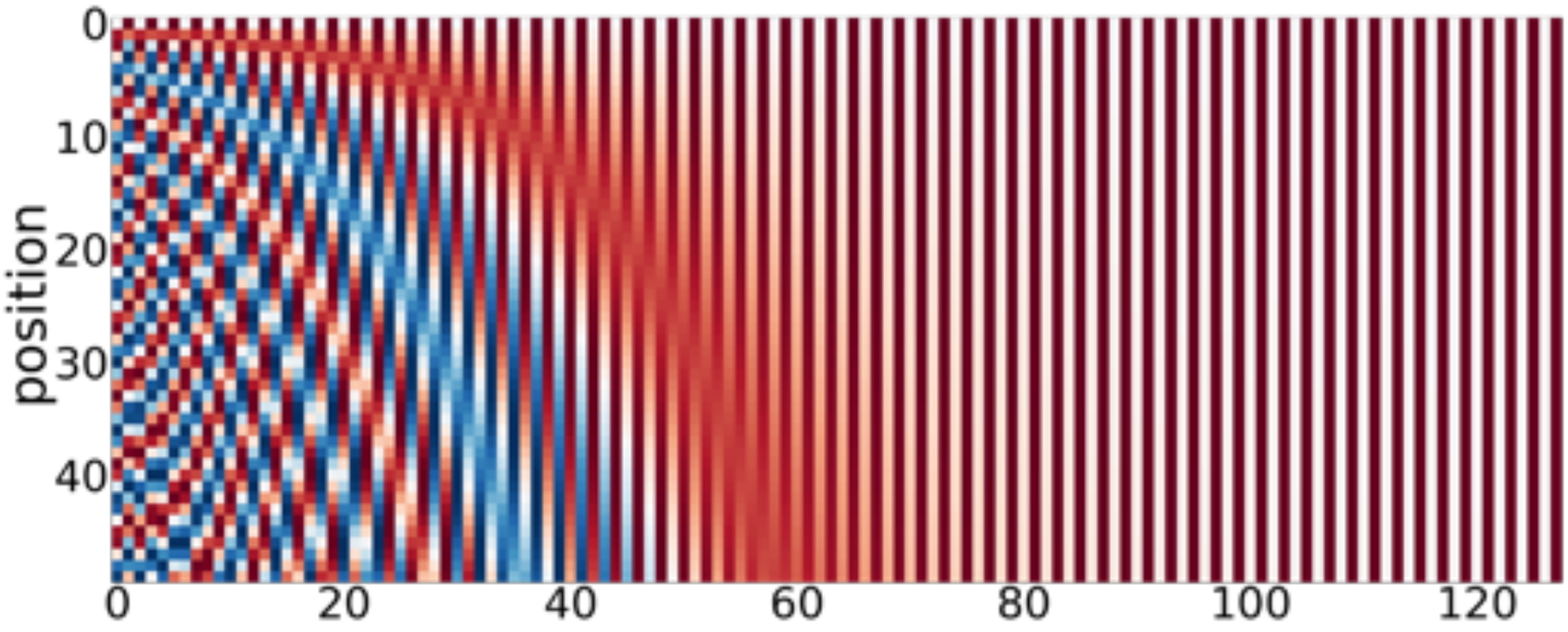
- **Contextualisation par auto-attention (bidirectionnelle)**

  - Concaténation de plusieurs têtes d'attention



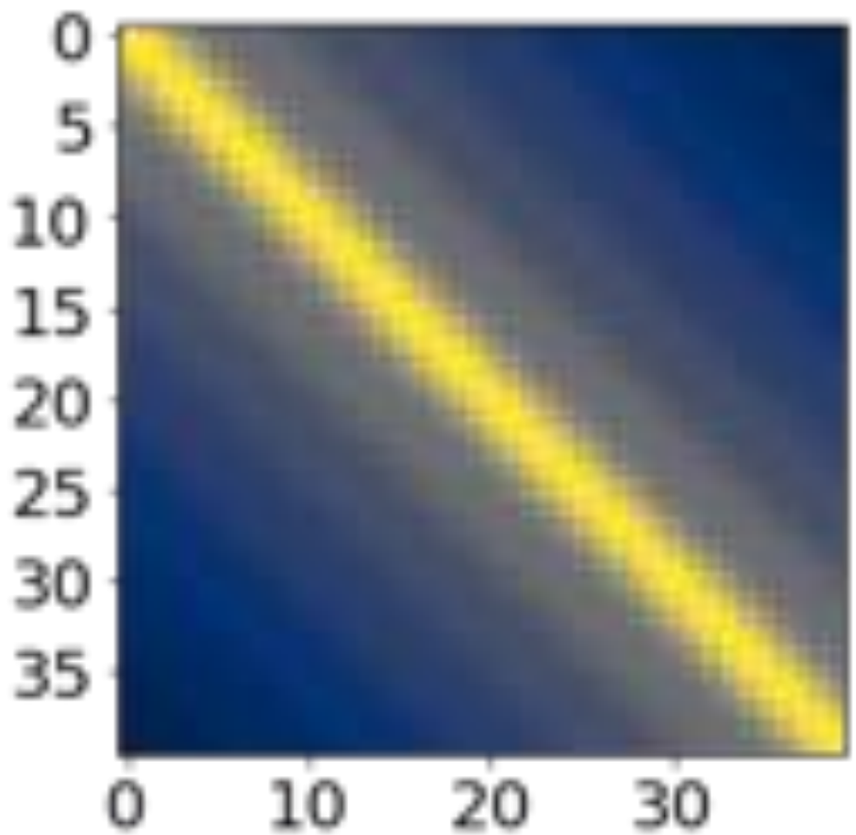
Encodage positionnel

- **Encodage sinusoïdale**
  - Vecteur u_i pour la position i : $u_{ij} = \begin{cases} \sin \frac{i}{10000^{\frac{j}{d}}} & \text{si } j \text{ est pair,} \\ \cos \frac{i}{10000^{\frac{j-1}{d}}} & \text{sinon.} \end{cases}$



Encodage positionnel

- **Position absolue et position relative**
  - Pour k fixé, lien linéaire entre u_{i+k} et u_i
  - Produit-scalaire « lisse »



Auto-attention par produit scalaire normalisé

- **Transformation en vecteurs-requêtes, -clés et -valeurs**

- Paramètres : $W_Q, W_K, W_V \in \mathbb{R}^{d \times d'}$

$$Q = XW_Q \quad , \quad K = XW_K \quad , \quad V = XV$$

- **Mesure des poids d'attention entre vecteurs-requêtes et -clés**

$$A = \underbrace{\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d'}}\right)}_{n \times n}$$

$$P_{ij} = \frac{q_i \cdot k_j}{\sqrt{d'}}$$

$$A = \mathrm{softmax}(P)$$

- **Contextualisation selon les vecteurs-valeurs**

$$C \in \mathbb{R}^{n \times d'} \qquad C = AV$$

Auto-attention multi-têtes

- **Concaténation**

$$Y \in \mathbb{R}^{n \times d}$$

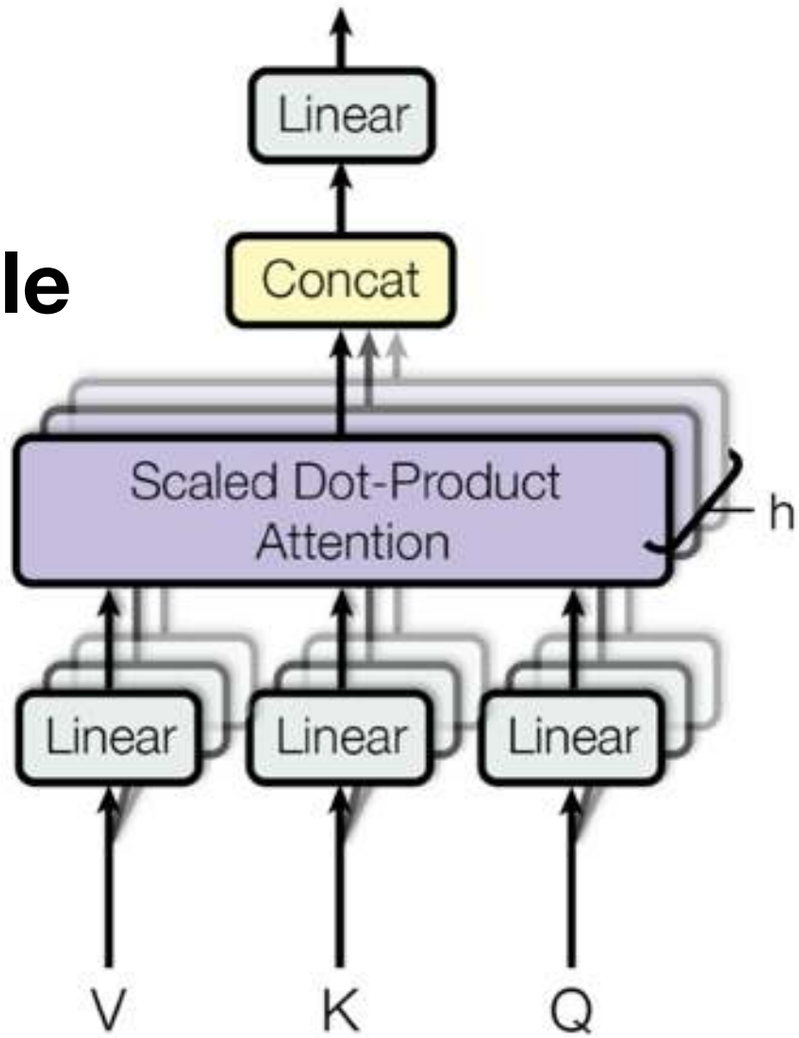
$$Y = C_1 \oplus C_2 \oplus \dots \oplus C_H$$

- **Transformation linéaire additionnelle**

$$Y' = Y W_0 \qquad W_0 \in \mathbb{R}^{d \times d}$$

- **Connexion résiduelle**

$$Y'' = X + Y'$$



Transformer

- **Détails importants**
    - Découpage du texte en entrée
        - Algorithme « Word Piece »
    - Normalisation entre les blocs
        - « Layer Normalization » (Preprint, Ba, Kiros & Hinton 2016)
- **Avantages**

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Modèles de langue pré-entraînés

BERT & GTP-x

- **GPT-x : décodeur Transformer profond**
  - *Language Models are Few-Shot Learners* (Preprint, Brown *et al.* 2020)
- **BERT : encodeur Transformer profond**
  - *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (Devlin et al. @ NAACL 2019)

~ 7,2 M x 12 + 23 M ~ 110 M params

BERT- base

• Représentations de mots : $U \in \mathbb{R}^{30\,000 \times 768}$
  • vocabulaire de 30 000 mots
  • $d = 768$

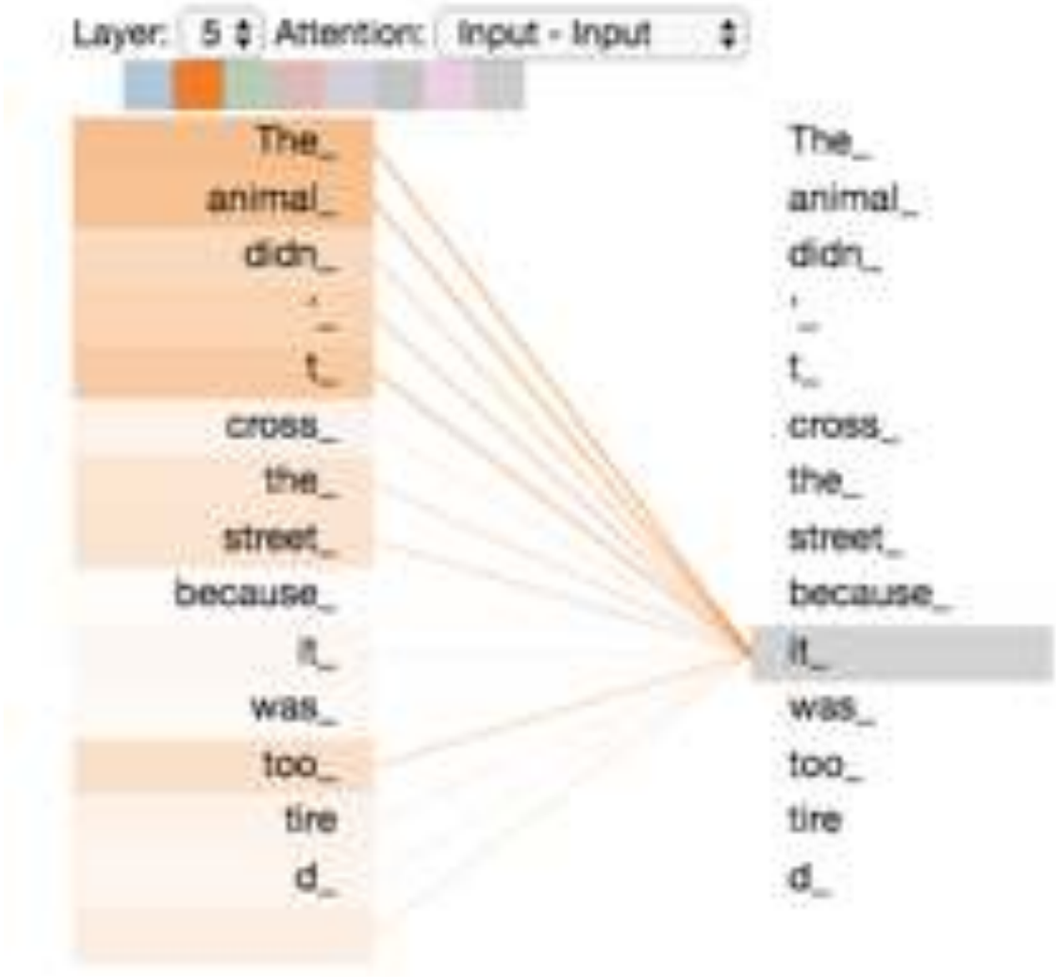
• transformer x12 ~ 7,2 M params
  • Auto-attention 12 têtes : • $W_Q, W_K, W_V \in \mathbb{R}^{768 \times 64}$ (768/12)
    • $W_O \in \mathbb{R}^{768 \times 768}$
    } $768 \times 64 \times 3 \times 12 + 768^2 \simeq 2{,}4$ M params
  • Réseau "feed forward" : • $W_1 \in \mathbb{R}^{768 \times 3072}$
    • $W_2 \in \mathbb{R}^{3072 \times 768}$
    } $3072 \times 768 \times 2 \simeq 4{,}8$ M params

BERT

- **Pré-entraînement général (très coûteux)**
  - « Masked Language Modeling » Il pleut, le temps est [MASK] aujourd'hui
  - « Next Sentence Prediction » [CLS] Il pleut...
- **Spécialisation (relativement peu coûteux)**
  - Ajout d'une dernière couche ad-hoc

BERT

- **Têtes d'attention spécialisées**
  - Résolution des coréférences, localisation des sujets, *etc.*
  - https://github.com/jessevig/bertviz



Distillation ad-hoc

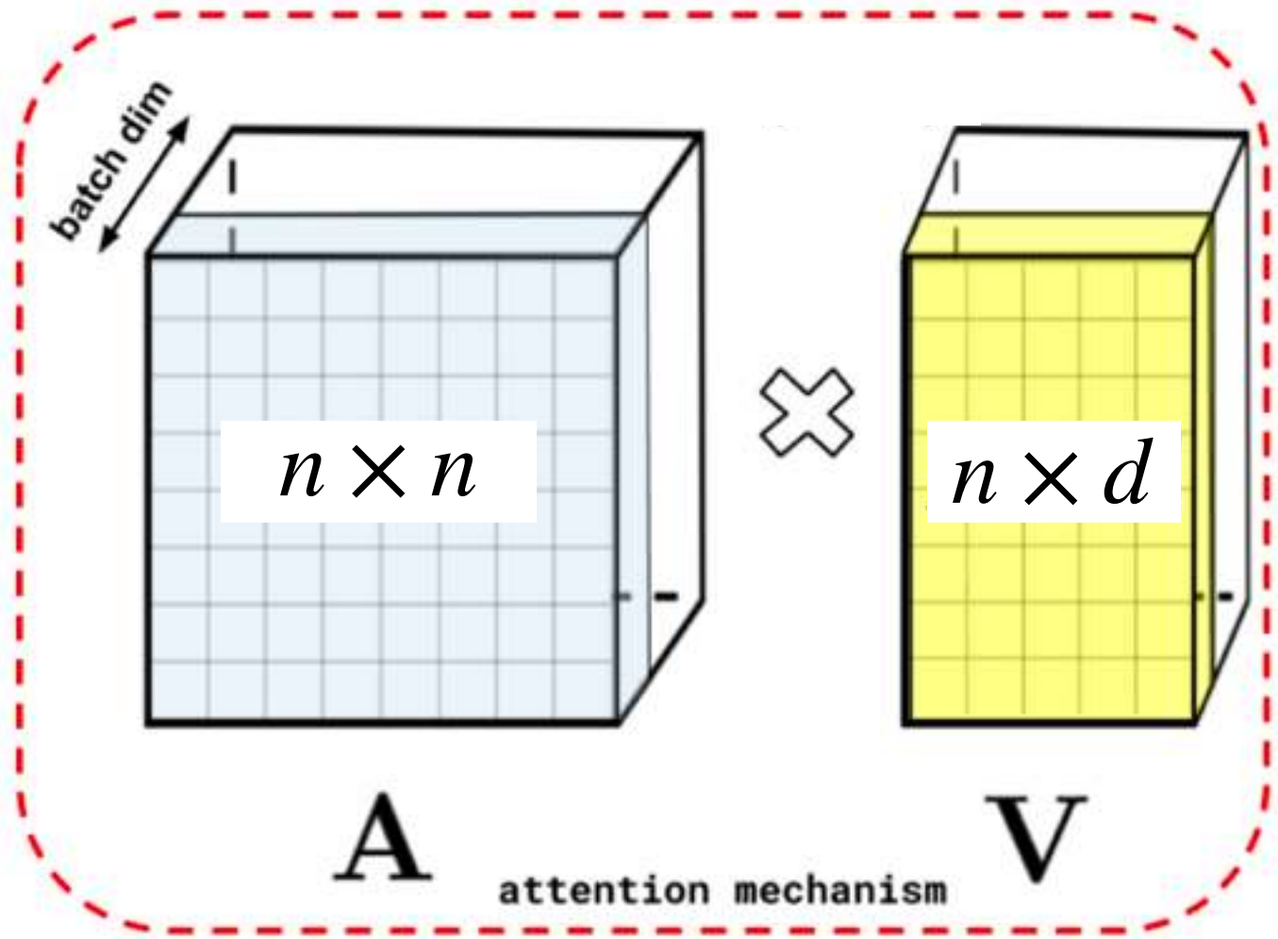
- **Réduction du coût d'utilisation**
  - *Rethinking complex neural network architectures for document classification* (Adhikari *et al.* @ NAACL 2019)
  - *Exploring the Limits of Simple Learners in Knowledge Distillation for Document Classification with DocBERT (Adhikari et al. @ RepL4NLPN workshop — ACL 2020)*

#	Model	Reuters		AAPD		IMDB		Yelp '14	
		Val. F_1	Test F_1	Val. F_1	Test F_1	Val. Acc.	Test Acc.	Val. Acc.	Test Acc.
1	LR	77.0	74.8	67.1	64.9	43.1	43.4	61.1	60.9
2	SVM	89.1	86.1	71.1	69.1	42.5	42.4	59.7	59.6
3	KimCNN	83.5 ±0.4	80.8 ±0.3	54.5 ±1.4	51.4 ±1.3	42.9 ±0.3	42.7 ±0.4	66.5 ±0.1	66.1 ±0.6
4	XML-CNN	88.8 ±0.5	86.2 ±0.3	70.2 ±0.7	68.7 ±0.4	–	–	–	–
5	HAN	87.6 ±0.5	85.2 ±0.6	70.2 ±0.2	68.0 ±0.6	51.8 ±0.3	51.2 ±0.3	68.2 ±0.1	67.9 ±0.1
6	SGM	82.5 ±0.4	78.8 ±0.9	–	71.0†	–	–	–	–
7	LSTM	89.1 ±0.8	87.0 ±0.5	73.1 ±0.4	70.5 ±0.5	53.4 ±0.2	52.8 ±0.3	69.0 ±0.1	68.7 ±0.1
8	$BERT_{base}$	90.5	89.0	75.3	73.4	54.4	54.2	72.1	72.0
9	$BERT_{large}$	**92.3**	**90.7**	**76.6**	**75.2**	**56.0**	**55.6**	**72.6**	**72.5**
10	KD-LSTM	91.0 ±0.2	88.9 ±0.2	75.4 ±0.2	72.9 ±0.3	54.5 ±0.1	53.7 ±0.3	69.7 ±0.1	69.4 ±0.1
11	KD-KimCNN	90.0 ±0.3	87.0 ±0.2	72.7 ±0.4	70.6 ±0.1	49.0 ±0.2	48.3 ±0.3	66.5 ±0.1	66.2 ±0.0
12	KD-LR	87.0	83.7	73.1	71.3	43.8	43.3	62.7	62.3

Réduction de la complexité de l’auto-attention

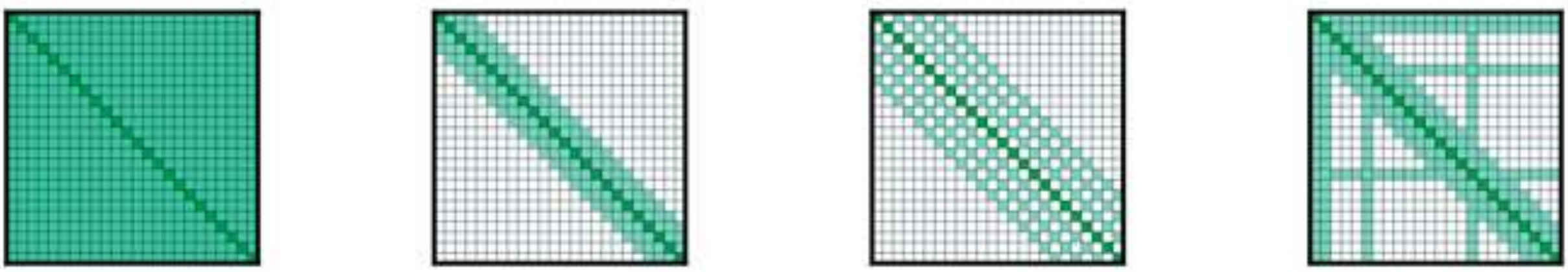
Transformer

- **Auto-attention « dense »**
  - Complexité en temps : $O(n^2)$



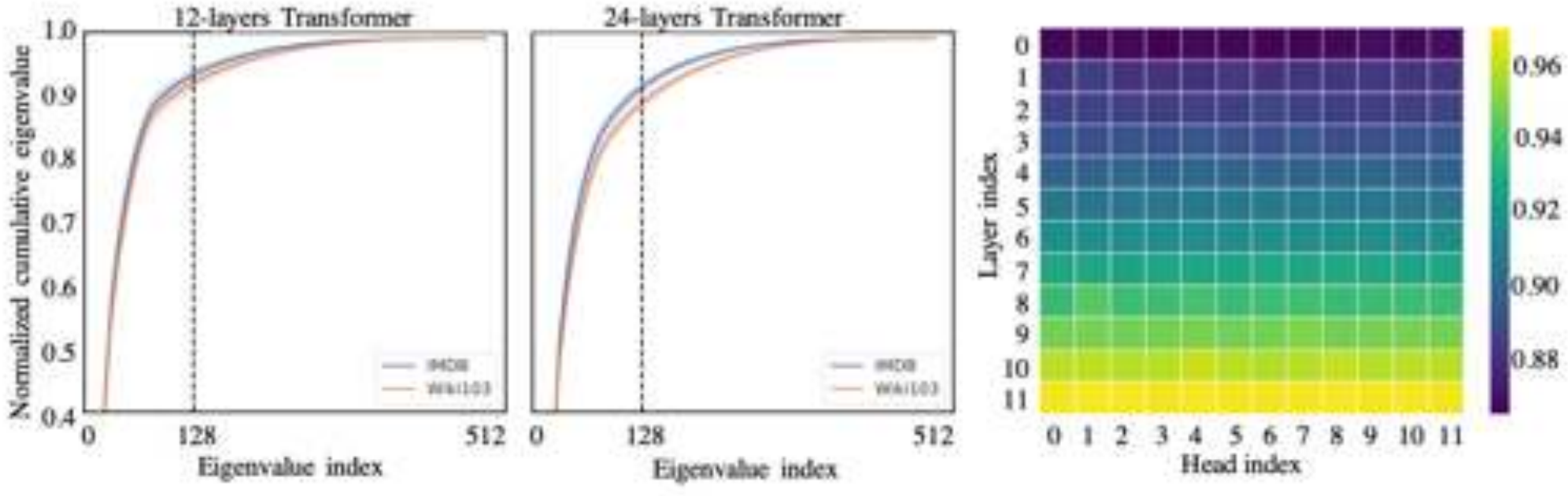
Sparse Transformer & LongFormer

- **Auto-attention « creuse »**
  - Complexité en temps
    - Sparse Transformer : $O(n\sqrt{n})$
    - LongFormer : $O(n \times (k + m))$



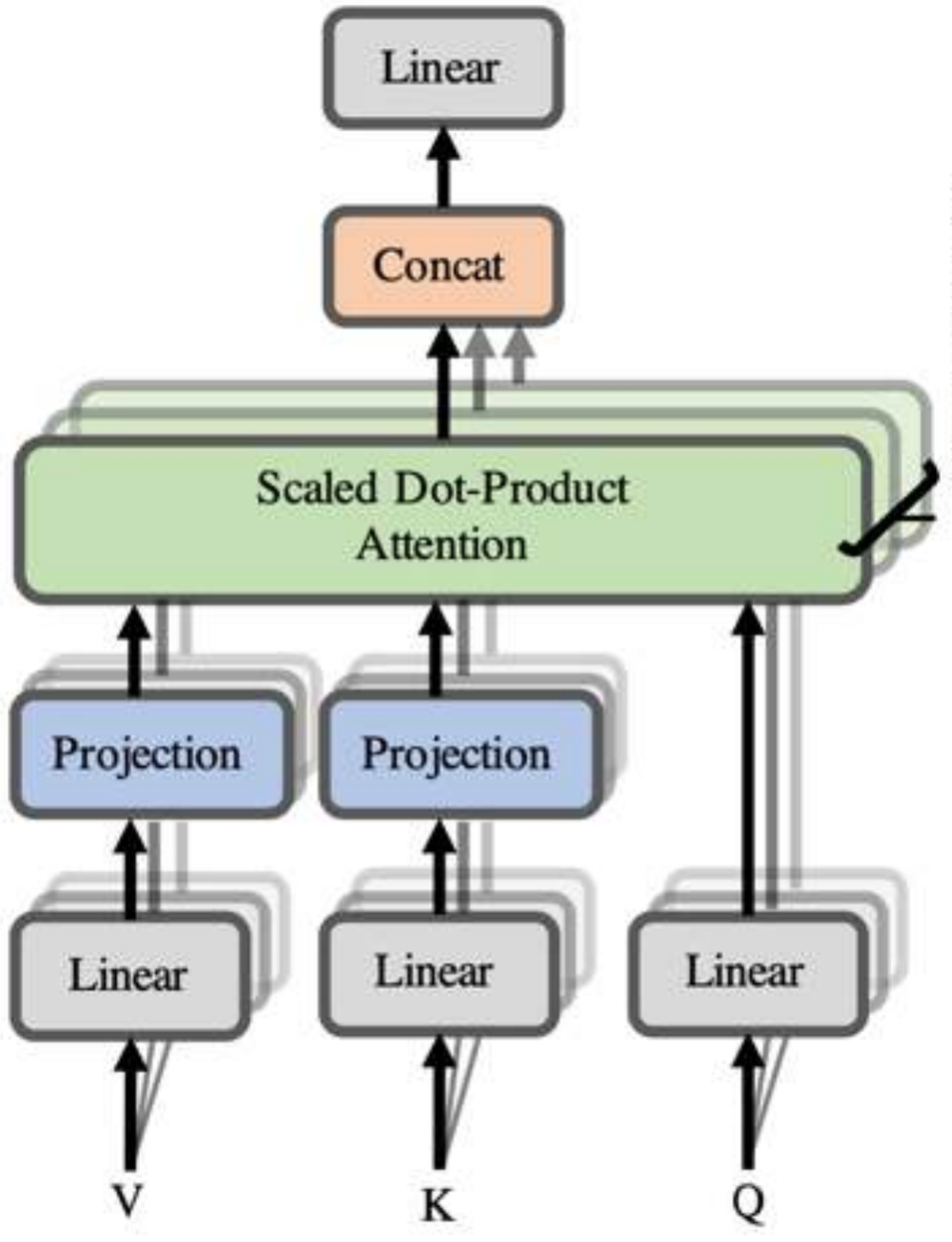
LinFormer

- **Auto-attention de rang faible**
    - *Linformer: Self-Attention with Linear Complexity* (Preprint, Wang *et al.* 2020)
- Complexité en temps : $O(n)$



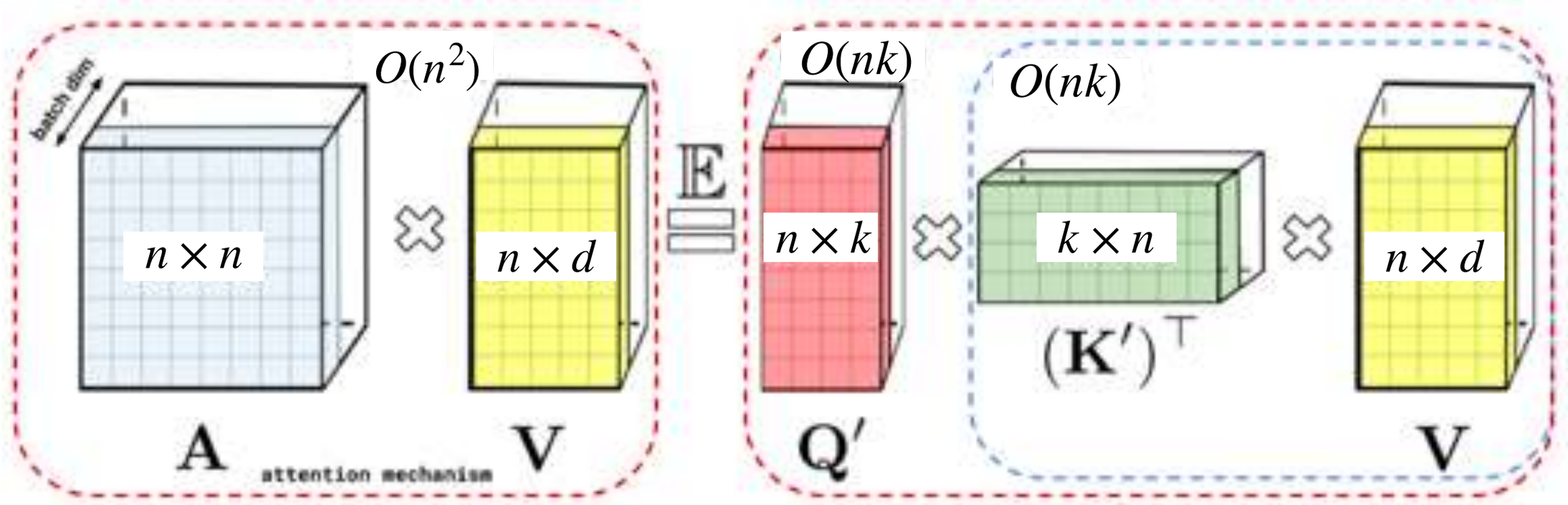
LinFormer

- **Auto-attention de rang faible**



Performer

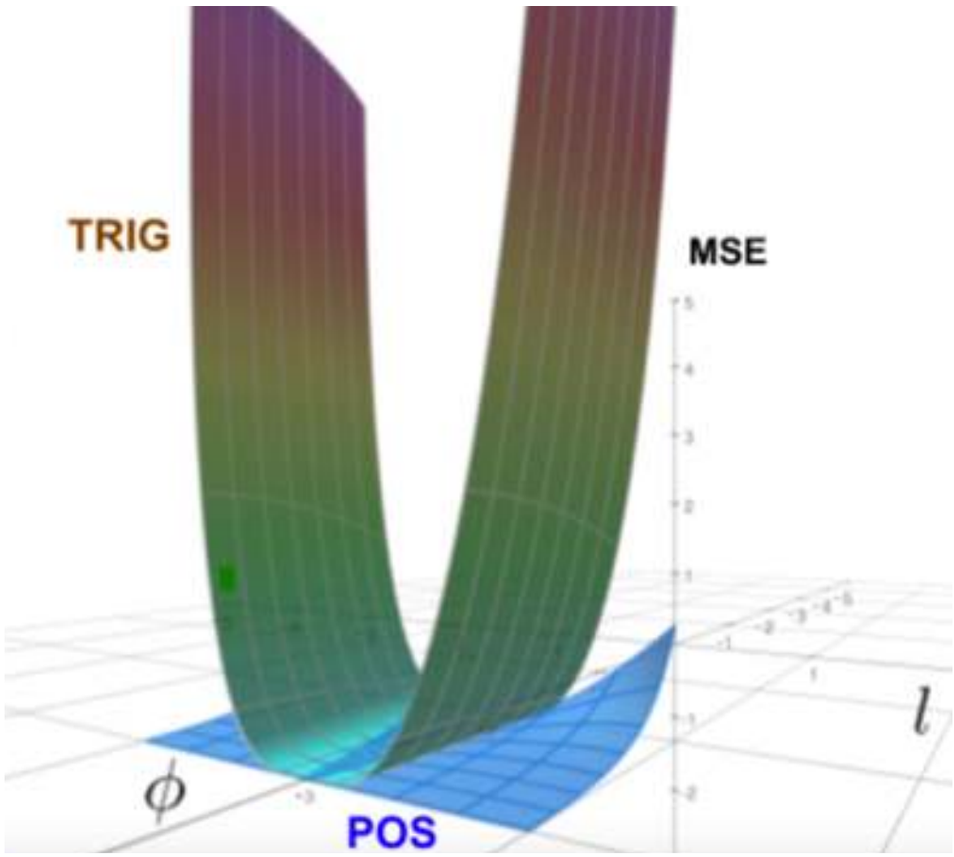
- **Approximation par noyau aléatoire**
  - *Rethinking attention with Performers* (Choromanski *et al.* @ ICLR 2021)
  - Complexité en temps : $O(n)$



$$\phi(\mathbf{x}) = \frac{h(\mathbf{x})}{\sqrt{m}}(f_1(\omega_1^\top \mathbf{x}), ..., f_1(\omega_m^\top \mathbf{x}), ..., f_l(\omega_1^\top \mathbf{x}), ..., f_l(\omega_m^\top \mathbf{x}))$$

Performer

- **Approximation par noyau aléatoire**
    - Fast Attention Via positive Orthogonal Random features (FAVOR+)



Perspectives

- **Distillation**
    - *TinyBERT* (EMNLP 2020), *MiniLM* (NeurIPS 2020), *FastBERT* (ACL 2020), *DistilBERT* ()…
- **Transfert cross-domaine**
    - *Pretrained Transformers as universal computation engines* (Preprint, Lu et al. 2021)