Contrôle périodique
INF8225 IA : techniques probabilistes et d'apprentissage

Toute documentation sur papier est autorisée.

All forms of documentation on paper are allowed.

Question	Points
Q1	19
Q2	5
Q3	10
Total	34

Hiver 2025, le 28 févr. 12h55 à 15h25.

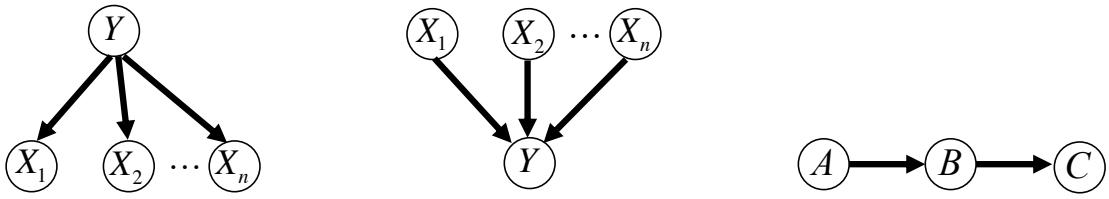

FIGURE 1 – Réseaux bayésiens où toutes les variables sont discrètes avec k états.

Question 1 (version française) (19 points)

Fournissez des équations et/ou des explications pour ces questions comme demandé ci-dessous.

a) (3 points) Écrivez une équation pour le modèle de probabilité que chaque réseau bayésien dans la figure ci-dessus représente. Assurez-vous que les factorisations soient claires.

(gauche) $P(Y, X_1, X_2, \ldots, X_n) = P(Y) \prod_{i=1}^{n} P(X_i \mid Y)$

(milieu) $P(Y, X_1, X_2, \ldots, X_n) = P(Y \mid X_1, X_2, \ldots, X_n) \prod_{i=1}^{n} P(X_i)$

(droite) $P(A, B, C) = P(A)P(B \mid A)P(C \mid B)$

b) (4 points) (i-iii) Donnez une expression pour le nombre de paramètres de chaque modèle en fonction de k et n et expliquez comment vous avez obtenu chaque expression. Puis, (iv) donnez une expression pour le nombre de paramètres libres pour le réseau du milieu uniquement.

i) Pour le réseau de gauche, nous avons k paramètres pour $P(Y)$ et pour chaque distribution conditionnelle nous avons k^2 paramètres, donc le total est $k + nk^2$.
ii) Pour le réseau bayésien du milieu, les probabilités inconditionnelles possèdent chacune k paramètres, ce qui donne nk paramètres. Pour la distribution conditionnelle de Y sachant les n X, nous avons k^{n+1} paramètres, donc le total est $nk + k^{n+1}$.
iii) Pour le réseau de droite, nous avons $k + 2k^2$ paramètres.

Nombre de paramètres libres dans chaque modèle

Supposons que toutes les variables ont k états.

Réseau de gauche

La distribution conjointe est

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y) \prod_{i=1}^{n} P(X_i \mid Y).$$

— $P(Y)$ est une distribution sur k états, donc elle nécessite $k - 1$ paramètres libres.
— Pour chaque i, $P(X_i \mid Y)$ est une distribution conditionnelle. Pour chacune des k valeurs de Y, il existe une distribution sur k états pour X_i, nécessitant $k - 1$ paramètres libres. Ainsi, chaque $P(X_i \mid Y)$ nécessite $k\,(k - 1)$ paramètres.

Puisqu'il y a n distributions conditionnelles, le nombre total de paramètres libres est

$$\boxed{(k-1) + n\,k\,(k-1) = (k-1)(1+nk).}$$

Réseau du milieu

La distribution conjointe est

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y \mid X_1, X_2, \ldots, X_n) \prod_{i=1}^{n} P(X_i).$$

- Chaque $P(X_i)$ est une distribution marginale sur k états et nécessite $k-1$ paramètres. Avec n distributions, cela apporte $n(k-1)$ paramètres.
- $P(Y \mid X_1, X_2, \ldots, X_n)$ est défini pour chaque configuration des n variables. Comme chaque X_i a k états, il y a k^n configurations. Pour chaque configuration, la distribution sur Y (avec k états) nécessite $k-1$ paramètres. Ainsi, cette partie nécessite $k^n(k-1)$ paramètres.

Donc, le nombre total de paramètres libres est

$$\boxed{n(k-1) + k^n(k-1) = (k-1)(n+k^n).}$$

Réseau de droite

La distribution conjointe est donnée par

$$P(A, B, C) = P(A)P(B \mid A)P(C \mid B).$$

- $P(A)$ est une distribution sur k états, nécessitant $k-1$ paramètres.
- $P(B \mid A)$ est défini pour chacune des k valeurs de A. Pour chaque valeur de A, la distribution sur B (avec k états) nécessite $k-1$ paramètres, soit au total $k(k-1)$ paramètres.
- $P(C \mid B)$ est défini pour chacune des k valeurs de B. De même, cela nécessite $k(k-1)$ paramètres.

Ainsi, le nombre total de paramètres libres est

$$\boxed{(k-1) + k(k-1) + k(k-1) = (k-1)(1+2k).}$$

c) (6 points) i) Comment calculeriez-vous $P(X_1 \mid X_2)$ pour le réseau de gauche ? ii) Comment calculeriez-vous $P(X_1 \mid X_2)$ pour le réseau du milieu ? iii) Comment calculeriez-vous $P(A \mid C)$ pour le réseau de droite ?

(i) Calcul de $P(X_1 \mid X_2)$ pour le réseau de gauche

La distribution conjointe du modèle est donnée par :

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y) \prod_{i=1}^{n} P(X_i \mid Y).$$

Étape 1 : Calcul de $P(X_1, X_2)$

Pour obtenir $P(X_1, X_2)$, nous devons marginaliser sur Y et sur les autres variables $X_3, X_4, \ldots, X_n$:

$$P(X_1, X_2) = \sum_{y} \sum_{x_3} \cdots \sum_{x_n} P(Y = y, X_1, X_2, x_3, \ldots, x_n).$$

En substituant la forme factorisée, nous avons :

$$P(X_1, X_2) = \sum_y P(Y = y)\, P(X_1 \mid Y = y)\, P(X_2 \mid Y = y) \prod_{i=3}^{n} \left(\sum_{x_i} P(x_i \mid Y = y) \right).$$

Puisque pour tout $i \geq 3$,

$$\sum_{x_i} P(x_i \mid Y = y) = 1,$$

cela se simplifie en :

$$P(X_1, X_2) = \sum_y P(Y = y)\, P(X_1 \mid Y = y)\, P(X_2 \mid Y = y).$$

Étape 2 : Calcul de $P(X_2)$

De même, pour obtenir $P(X_2)$, nous marginalisons sur Y et toutes les variables sauf X_2 :

$$P(X_2) = \sum_y \sum_{x_1} \sum_{x_3} \cdots \sum_{x_n} P(Y = y, X_1, X_2, x_3, \ldots, x_n).$$

En utilisant la factorisation, ceci devient :

$$P(X_2) = \sum_y P(Y = y)\, P(X_2 \mid Y = y) \prod_{\substack{i=1 \\ i \neq 2}}^{n} \left(\sum_{x_i} P(x_i \mid Y = y) \right).$$

Encore, puisque $\sum_{x_i} P(x_i \mid Y = y) = 1$ pour $i \neq 2$, nous avons :

$$P(X_2) = \sum_y P(Y = y)\, P(X_2 \mid Y = y).$$

Étape 3 : Calcul de $P(X_1 \mid X_2)$

Enfin, par définition de la probabilité conditionnelle :

$$P(X_1 \mid X_2) = \frac{P(X_1, X_2)}{P(X_2)} = \frac{\sum_y P(Y = y)\, P(X_1 \mid Y = y)\, P(X_2 \mid Y = y)}{\sum_y P(Y = y)\, P(X_2 \mid Y = y)}.$$

Ainsi, l'expression finale est :

$$\boxed{P(X_1 \mid X_2) = \frac{\sum_y P(Y = y)\, P(X_1 \mid Y = y)\, P(X_2 \mid Y = y)}{\sum_y P(Y = y)\, P(X_2 \mid Y = y)}}.$$

(ii) Pour le réseau du milieu :

La distribution conjointe est

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y \mid X_1, X_2, \ldots, X_n) \prod_{i=1}^{n} P(X_i).$$

Remarquez que lorsque nous marginalisons sur Y, la distribution conjointe des X_i devient

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i).$$

En particulier, X_1 et X_2 sont indépendants marginalement. Par conséquent,

$$\boxed{P(X_1 \mid X_2) = P(X_1)}.$$

—

(iii) Calcul de $P(A \mid C)$ pour le réseau de droite :

La distribution conjointe est

$$P(A, B, C) = P(A)P(B \mid A)P(C \mid B).$$

Pour calculer $P(A \mid C)$, marginalisez d'abord B :

$$P(A, C) = \sum_b P(A, B = b, C) = \sum_b P(A)P(B = b \mid A)P(C \mid B = b).$$

Ensuite, en utilisant la définition de la probabilité conditionnelle,

$$P(A \mid C) = \frac{P(A, C)}{P(C)} = \frac{\sum_b P(A)P(B = b \mid A)P(C \mid B = b)}{\sum_a \sum_b P(a)P(B = b \mid a)P(C \mid B = b)}.$$

Ainsi, la réponse est

$$\boxed{P(A \mid C) = \frac{\sum_b P(A)P(B = b \mid A)P(C \mid B = b)}{\sum_a \sum_b P(a)P(B = b \mid a)P(C \mid B = b)}}.$$

— Voici les expressions requises pour $P(X_1 \mid X_2)$ dans les réseaux de gauche et du milieu, et pour $P(A \mid C)$ dans le réseau de droite.

d) (3 points) i) Comment calculeriez-vous $P(X_1 \mid \text{do}(X_2 = x_2))$ pour le réseau de gauche ? ii) Comment calculeriez-vous $P(X_1 \mid \text{do}(X_2 = x_2))$ pour le réseau du milieu ? iii) Comment calculeriez-vous $P(A \mid \text{do}(C = c))$ pour le réseau de droite ?

Distributions interventionnelles

Rappel : l'opérateur do correspond à intervenir dans le modèle, ce qui revient à supprimer les flèches entrantes vers la variable concernée et à la fixer à une constante.

(i) Réseau de gauche

Le réseau de gauche a la distribution conjointe

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y) \prod_{i=1}^{n} P(X_i \mid Y).$$

Lorsque nous effectuons l'intervention $\text{do}(X_2 = x_2)$, nous supprimons la flèche de Y vers X_2 et fixons X_2 à x_2. La factorisation modifiée (tronquée) devient :

$$P_{\text{do}(X_2=x_2)}(Y, X_1, X_2, \ldots, X_n) = P(Y)\, P(X_1 \mid Y)\, \delta_{x_2}(X_2) \prod_{i=3}^{n} P(X_i \mid Y),$$

où $\delta_{x_2}(X_2)$ est l'indicatrice (ou delta de Dirac) indiquant que X_2 est fixé à x_2. Pour calculer $P(X_1 \mid \text{do}(X_2 = x_2))$, commencez par calculer la distribution interventionnelle conjointe de (X_1, X_2) en marginalisant sur Y (et sur les autres X_i, dont l'intégrale vaut 1) :

$$P_{\text{do}(X_2=x_2)}(X_1, X_2 = x_2) = \sum_y P(Y = y)\, P(X_1 \mid Y = y)\, P(X_2 = x_2 \mid \text{do}(X_2 = x_2)).$$

Puisque sous l'intervention $P(X_2 = x_2 \mid \text{do}(X_2 = x_2)) = 1$, nous avons

$$P_{\text{do}(X_2=x_2)}(X_1, X_2 = x_2) = \sum_y P(Y = y)\, P(X_1 \mid Y = y).$$

De même, la marginale pour X_2 est

$$P_{\text{do}(X_2=x_2)}(X_2 = x_2) = 1.$$

Ainsi,

$$\boxed{P(X_1 \mid \text{do}(X_2 = x_2)) = \sum_y P(X_1 \mid Y = y)\, P(Y = y).}$$

(ii) Réseau du milieu

Le réseau du milieu est défini par

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y \mid X_1, X_2, \ldots, X_n) \prod_{i=1}^{n} P(X_i).$$

Ici, toutes les X_i sont indépendantes marginalement. L'intervention $\text{do}(X_2 = x_2)$ fixe X_2 à x_2 sans modifier les distributions marginales des autres X_i. Par conséquent, sous l'intervention,

$$P_{\text{do}(X_2=x_2)}(X_1) = P(X_1).$$

Donc,

$$\boxed{P(X_1 \mid \text{do}(X_2 = x_2)) = P(X_1).}$$

(iii) Réseau de droite

Le réseau de droite a la factorisation

$$P(A, B, C) = P(A)P(B \mid A)P(C \mid B).$$

Intervenir sur C avec $\text{do}(C = c)$ supprime la dépendance de C à B et fixe C à c. La factorisation tronquée devient :

$$P_{\text{do}(C=c)}(A, B, C) = P(A)P(B \mid A)\, \delta_c(C).$$

Puisque l'intervention se situe en aval de A (c'est-à-dire que A est un ancêtre de C), la distribution de A reste inchangée :

$$\boxed{P(A \mid \text{do}(C = c)) = P(A).}$$

e) (2 points) Expliquez comment on pourrait paramétrer la distribution conditionnelle dans le réseau du milieu avec un modèle linéaire suivi d'une fonction softmax pour prédire la variable discrète Y. Encodez chaque X sous forme de vecteur one-hot. Expliquez mathématiquement comment ce modèle serait construit et comment la fonction de perte pour un tel modèle est définie et formulée.

Paramétrage de $P(Y \mid X_1, \ldots, X_n)$ avec un modèle linéaire et softmax

Supposons que chaque variable X_i prend l'une des k valeurs discrètes. Nous encodons d'abord chaque X_i sous forme de vecteur one-hot :

$$\mathbf{x}_i \in \{0, 1\}^k,$$

de sorte que si $X_i = j$ alors la $j$ème composante de $\mathbf{x}_i$ est 1 et les autres sont 0. Nous formons ensuite un vecteur d'entrée unique en concaténant les vecteurs one-hot de toutes les n variables :

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} \in \mathbb{R}^{nk}.$$

Ensuite, nous définissons une transformation linéaire qui mappe $\mathbf{x}$ vers un vecteur de *logits* pour Y :

$$\mathbf{z} = W\mathbf{x} + \mathbf{b},$$

où

$$W \in \mathbb{R}^{k\times(nk)} \quad \text{et} \quad \mathbf{b} \in \mathbb{R}^k.$$

Chaque composante z_j de $\mathbf{z}$ correspond au log-probabilité non normalisé de $Y = j$. Nous appliquons ensuite la fonction softmax pour obtenir une distribution de probabilité appropriée sur les k états de Y :

$$P(Y = j \mid X_1, \ldots, X_n) = \frac{\exp(z_j)}{\sum_{j'=1}^{k} \exp(z_{j'})}, \quad j = 1, \ldots, k.$$

Fonction de perte

Soit le jeu de données d'entraînement $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, où pour chaque exemple $\mathbf{x}^{(i)}$ est le vecteur one-hot concaténé pour $X_1, \ldots, X_n$ et $y^{(i)} \in \{1, \ldots, k\}$ est la véritable étiquette de Y. Pour un exemple, la log-vraisemblance négative (perte d'entropie croisée) est définie par :

$$L^{(i)} = -\log P(Y = y^{(i)} \mid \mathbf{x}^{(i)}).$$

En substituant l'expression softmax, nous obtenons :

$$L^{(i)} = -\log\left(\frac{\exp(z^{(i)}_{y^{(i)}})}{\sum_{j=1}^{k} \exp(z^{(i)}_j)}\right) = -z^{(i)}_{y^{(i)}} + \log\left(\sum_{j=1}^{k} \exp(z^{(i)}_j)\right).$$

La perte globale pour le jeu de données est la perte moyenne :

$$L = \frac{1}{N}\sum_{i=1}^{N} L^{(i)} = -\frac{1}{N}\sum_{i=1}^{N} \log P(Y = y^{(i)} \mid \mathbf{x}^{(i)}).$$

Cette fonction de perte est minimisée par rapport aux paramètres W et $\mathbf{b}$ lors de l'entraînement, ce qui ajuste le modèle linéaire afin que la distribution prédite $P(Y \mid X_1, \ldots, X_n)$ corresponde au mieux à la distribution réelle de Y compte tenu des entrées.

f) (1 point) Expliquez combien de paramètres seraient nécessaires pour paramétrer le modèle de (e) en fonction de n et k.

Nombre de paramètres pour le modèle linéaire-softmax

Rappelons que chaque variable X_i est encodée en one-hot comme un vecteur dans $\mathbb{R}^k$. En concaténant n de tels vecteurs, le vecteur d'entrée devient

$$\mathbf{x} \in \mathbb{R}^{nk}.$$

Le modèle linéaire est défini par :

$$\mathbf{z} = W\mathbf{x} + \mathbf{b},$$

où

$$W \in \mathbb{R}^{k\times(nk)} \quad \text{et} \quad \mathbf{b} \in \mathbb{R}^k.$$

Nombre de paramètres

— La matrice de poids W a k lignes et nk colonnes, ce qui donne :

$$k \times (nk) = nk^2 \quad \text{paramètres.}$$

— Le vecteur de biais $\mathbf{b}$ comporte k éléments, ajoutant :

$$k \quad \text{paramètres.}$$

Ainsi, le nombre total de paramètres est :

$$\boxed{nk^2 + k.}$$

Question 2 (version française) (5 points)

Donnez des réponses brèves pour cette section. Seules quelques lignes d'explications sont nécessaires.

a) (1 point) Quel type de fonctions d'activation a été utilisé pour les couches intermédiaires de la méthode YOLO originale et pourquoi ?

b) (2 points) Pourquoi un U-Net possède-t-il des connexions de saut (skip connections) et quelle est la motivation derrière la structure en "U" de l'architecture ?

c) (2 points) Dans vos propres mots, pourquoi la méthode ADAM est-elle si populaire ?

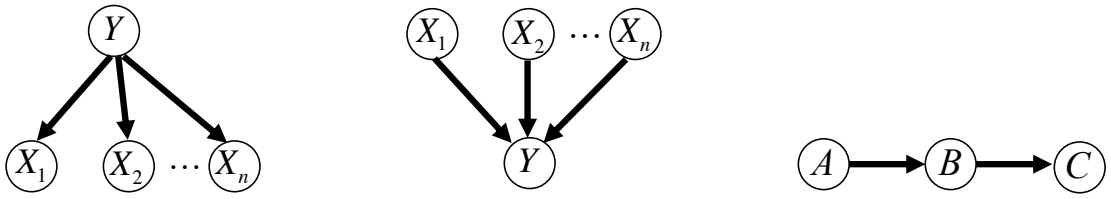

FIGURE 2 – Bayesian Networks where all variables are discrete with k states.

Question 1 (English version) (19 points)

Provide equations and/or explanations for these questions as requested below.

a) (3 points) Write an equation for the probability model that each Bayesian network in the figure above represents. Make sure that the factorizations are clear.

(left) $P(Y, X_1, X_2 \ldots, X_n) = P(Y) \prod_{i=1}^{N} P(X_i|Y)$

(middle) $P(Y, X_1, X_2 \ldots, X_n) = P(Y|X_1, X_2 \ldots, X_n) \prod_{i=1}^{N} P(X_i)$

(right) $P(A, B, C) = P(A)P(B|A)P(C|B)$

b) (4 points) (i-iii) Provide an expression for the number of parameters for each model as a function of k and n and explain how you obtained each expression. Then, (iv) provide an expression for the number of free parameters for just the middle network.

i) For the left network we have k parameters for $P(Y)$ and for each conditional we have k^2 parameters, so we have a total of $k + nk^2$.
ii) For the Bayesian Network in the middle, the unconditional probabilities each have k parameters, so there are nk parameters. For the conditional distribution of Y given the n Xs we have k^{n+1}, so we have a total of $nk + k^{n+1}$ parameters.
iii) For the network on the right we have $k + 2 * k^2$ parameters.

Number of Free Parameters in Each Model

Assume that all variables have k states.

Left Network

The joint distribution is

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y) \prod_{i=1}^{n} P(X_i \mid Y).$$

— $P(Y)$ is a distribution over k states, so it requires $k - 1$ free parameters.
— For each i, $P(X_i \mid Y)$ is a conditional distribution. For each of the k values of Y, there is a distribution over k states for X_i, which requires $k - 1$ free parameters. Thus, each $P(X_i \mid Y)$ requires $k\,(k - 1)$ parameters.

Since there are n such conditional distributions, the total number of free parameters is

$$\boxed{(k-1) + n\,k\,(k-1) = (k-1)(1+nk).}$$

Middle Network

The joint distribution is

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y \mid X_1, X_2, \ldots, X_n) \prod_{i=1}^{n} P(X_i).$$

— Each $P(X_i)$ is a marginal distribution over k states and requires $k-1$ parameters. With n such distributions, this contributes $n(k-1)$ parameters.
— $P(Y \mid X_1, X_2, \ldots, X_n)$ is defined for every configuration of the n variables. Since each X_i has k states, there are k^n configurations. For each configuration, the distribution over Y (with k states) requires $k-1$ parameters. Hence, this part requires $k^n(k-1)$ parameters.

Thus, the total number of free parameters is

$$\boxed{n(k-1) + k^n(k-1) = (k-1)(n+k^n).}$$

Right Network

The joint distribution is given by

$$P(A, B, C) = P(A)P(B \mid A)P(C \mid B).$$

— $P(A)$ is a distribution over k states, requiring $k-1$ parameters.
— $P(B \mid A)$ is defined for each of the k states of A. For each state of A, the distribution over B (with k states) requires $k-1$ parameters, so in total $k(k-1)$ parameters.
— $P(C \mid B)$ is defined for each of the k states of B. Similarly, this requires $k(k-1)$ parameters.

Thus, the total number of free parameters is

$$\boxed{(k-1) + k(k-1) + k(k-1) = (k-1)(1+2k).}$$

c) (6 points) i) How would you compute $P(X_1|X_2)$ for the network on the left ? ii) How would you compute $P(X_1|X_2)$ for the middle network ? iii) How would you compute $P(A|C)$ for the network on the right ?

(i) Computing $P(X_1 \mid X_2)$ for the Left Network

The joint distribution for the model is given by :

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y) \prod_{i=1}^{n} P(X_i \mid Y).$$

Step 1 : Compute $P(X_1, X_2)$

To obtain $P(X_1, X_2)$, we must marginalize over Y and the other variables $X_3, X_4, \ldots, X_n$:

$$P(X_1, X_2) = \sum_{y} \sum_{x_3} \cdots \sum_{x_n} P(Y = y, X_1, X_2, x_3, \ldots, x_n).$$

Substituting the factorized form, we have :

$$P(X_1, X_2) = \sum_{y} P(Y = y)\, P(X_1 \mid Y = y)\, P(X_2 \mid Y = y) \prod_{i=3}^{n} \left(\sum_{x_i} P(x_i \mid Y = y) \right).$$

Since for every $i \geq 3$,

$$\sum_{x_i} P(x_i \mid Y = y) = 1,$$

this simplifies to :

$$P(X_1, X_2) = \sum_y P(Y = y)\, P(X_1 \mid Y = y)\, P(X_2 \mid Y = y).$$

Step 2 : Compute $P(X_2)$

Similarly, to obtain $P(X_2)$, we marginalize over Y and all variables except X_2 :

$$P(X_2) = \sum_y \sum_{x_1} \sum_{x_3} \cdots \sum_{x_n} P(Y = y, X_1, X_2, x_3, \ldots, x_n).$$

Using the factorization, this becomes :

$$P(X_2) = \sum_y P(Y = y)\, P(X_2 \mid Y = y) \prod_{\substack{i=1 \\ i \neq 2}}^{n} \left(\sum_{x_i} P(x_i \mid Y = y) \right).$$

Again, since $\sum_{x_i} P(x_i \mid Y = y) = 1$ for $i \neq 2$, we have :

$$P(X_2) = \sum_y P(Y = y)\, P(X_2 \mid Y = y).$$

Step 3 : Compute $P(X_1 \mid X_2)$

Finally, by the definition of conditional probability :

$$P(X_1 \mid X_2) = \frac{P(X_1, X_2)}{P(X_2)} = \frac{\sum_y P(Y = y)\, P(X_1 \mid Y = y)\, P(X_2 \mid Y = y)}{\sum_y P(Y = y)\, P(X_2 \mid Y = y)}.$$

Thus, the final expression is :

$$\boxed{P(X_1 \mid X_2) = \frac{\sum_y P(Y = y)\, P(X_1 \mid Y = y)\, P(X_2 \mid Y = y)}{\sum_y P(Y = y)\, P(X_2 \mid Y = y)}}.$$

(ii) For the middle network :

The joint distribution is

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y|X_1, X_2, \ldots, X_n) \prod_{i=1}^{n} P(X_i).$$

Notice that when we marginalize over Y, the joint distribution of the X_i's becomes

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i).$$

In particular, X_1 and X_2 are marginally independent. Therefore,

$$\boxed{P(X_1|X_2) = P(X_1)}.$$

—

(ii) Computing $P(A|C)$ for the right network :

The joint distribution is

$$P(A, B, C) = P(A)P(B|A)P(C|B).$$

To compute $P(A|C)$, first marginalize out B :

$$P(A, C) = \sum_b P(A, B = b, C) = \sum_b P(A)P(B = b|A)P(C|B = b).$$

Then, using the definition of conditional probability,

$$P(A|C) = \frac{P(A, C)}{P(C)} = \frac{\sum_b P(A)P(B = b|A)P(C|B = b)}{\sum_a \sum_b P(a)P(B = b|a)P(C|B = b)}.$$

Thus, the answer is

$$\boxed{P(A|C) = \frac{\sum_b P(A)P(B = b|A)P(C|B = b)}{\sum_a \sum_b P(a)P(B = b|a)P(C|B = b)}}.$$

— These are the required expressions for $P(X_1|X_2)$ in the left and middle networks, and for $P(A|C)$ in the right network.

d) (3 points) i) How would you compute $P(X_1|\text{do}(X_2 = x_2))$ for the network on the left ? ii) How would you compute $P(X_1|\text{do}(X_2 = x_2))$ for the middle network ? iii) How would you compute $P(A|do(C = c))$ for the network on the right ?

Interventional Distributions

Recall that the do-operator corresponds to intervening in the model, which amounts to removing the incoming edges to the intervened variable and fixing it to a constant.

(i) Left Network

The left network has the joint

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y) \prod_{i=1}^{n} P(X_i \mid Y).$$

When we perform the intervention $\text{do}(X_2 = x_2)$, we remove the arrow from Y to X_2 and set X_2 to x_2. The modified (truncated) factorization becomes :

$$P_{\text{do}(X_2=x_2)}(Y, X_1, X_2, \ldots, X_n) = P(Y)\, P(X_1 \mid Y)\, \delta_{x_2}(X_2) \prod_{i=3}^{n} P(X_i \mid Y),$$

where $\delta_{x_2}(X_2)$ is the indicator (or Dirac delta) that X_2 is fixed at x_2. To compute $P(X_1 \mid \text{do}(X_2 = x_2))$, first compute the joint interventional distribution of (X_1, X_2) by marginalizing over Y (and over the other X_i's, which integrate to 1) :

$$P_{\text{do}(X_2=x_2)}(X_1, X_2 = x_2) = \sum_y P(Y = y)\, P(X_1 \mid Y = y)\, P(X_2 = x_2 \mid \text{do}(X_2 = x_2)).$$

Since under the intervention $P(X_2 = x_2 \mid \text{do}(X_2 = x_2)) = 1$, we have

$$P_{\text{do}(X_2=x_2)}(X_1, X_2 = x_2) = \sum_y P(Y = y)\, P(X_1 \mid Y = y).$$

Similarly, the marginal for X_2 is

$$P_{\text{do}(X_2=x_2)}(X_2 = x_2) = 1.$$

Thus,

$$\boxed{P(X_1 \mid \text{do}(X_2 = x_2)) = \sum_y P(X_1 \mid Y = y)\, P(Y = y).}$$

(ii) Middle Network

The middle network is defined by

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y \mid X_1, X_2, \ldots, X_n) \prod_{i=1}^{n} P(X_i).$$

Here all the X_i's are marginally independent. The intervention $\text{do}(X_2 = x_2)$ fixes X_2 at x_2 but does not alter the marginal distributions of the other X_i's. Hence, under the intervention,

$$P_{\text{do}(X_2=x_2)}(X_1) = P(X_1).$$

Therefore,

$$\boxed{P(X_1 \mid \text{do}(X_2 = x_2)) = P(X_1).}$$

(iii) Right Network

The right network has the factorization

$$P(A, B, C) = P(A)P(B \mid A)P(C \mid B).$$

Intervening on C with $\text{do}(C = c)$ removes the dependency of C on B and sets C to c. The truncated factorization becomes :

$$P_{\text{do}(C=c)}(A, B, C) = P(A)P(B \mid A)\, \delta_c(C).$$

Since the intervention is downstream of A (i.e. A is an ancestor of C), the distribution of A remains unaffected :

$$\boxed{P(A \mid \text{do}(C = c)) = P(A).}$$

e) (2 points) Explain how one could parameterize the conditional distribution in the middle network with a linear model followed by a softmax to predict the discrete variable Y. Encode each X as a one hot vector. Explain how this model would be constructed with mathematics and explain how the loss for such a model is defined and written.

Parameterizing $P(Y \mid X_1, \ldots, X_n)$ with a Linear Model and Softmax

Assume that each variable X_i takes one of k discrete states. We first encode each X_i as a one-hot vector :

$$\mathbf{x}_i \in \{0, 1\}^k,$$

so that if $X_i = j$ then the jth component of $\mathbf{x}_i$ is 1 and the others are 0. We then form a single input vector by concatenating the one-hot vectors for all n variables :

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} \in \mathbb{R}^{nk}.$$

Next, we define a linear transformation that maps $\mathbf{x}$ to a vector of *logits* for Y :

$$\mathbf{z} = W\mathbf{x} + \mathbf{b},$$

where

$$W \in \mathbb{R}^{k\times(nk)} \quad \text{and} \quad \mathbf{b} \in \mathbb{R}^k.$$

Each component z_j of $\mathbf{z}$ corresponds to the unnormalized log-probability of $Y = j$. We then apply the softmax function to obtain a proper probability distribution over the k states of Y :

$$P(Y = j \mid X_1, \ldots, X_n) = \frac{\exp(z_j)}{\sum_{j'=1}^{k} \exp(z_{j'})}, \quad j = 1, \ldots, k.$$

Loss Function

Let the training dataset be $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, where for each example $\mathbf{x}^{(i)}$ is the concatenated one-hot vector for $X_1, \ldots, X_n$ and $y^{(i)} \in \{1, \ldots, k\}$ is the true label for Y. For a single example, the negative log-likelihood (cross-entropy loss) is defined as :

$$L^{(i)} = -\log P(Y = y^{(i)} \mid \mathbf{x}^{(i)}).$$

Substituting the softmax expression, we have :

$$L^{(i)} = -\log\left(\frac{\exp(z^{(i)}_{y^{(i)}})}{\sum_{j=1}^{k} \exp(z^{(i)}_j)}\right) = -z^{(i)}_{y^{(i)}} + \log\left(\sum_{j=1}^{k} \exp(z^{(i)}_j)\right).$$

The overall loss for the dataset is the average loss :

$$L = \frac{1}{N}\sum_{i=1}^{N} L^{(i)} = -\frac{1}{N}\sum_{i=1}^{N} \log P(Y = y^{(i)} \mid \mathbf{x}^{(i)}).$$

This loss function is minimized with respect to the parameters W and $\mathbf{b}$ during training, which adjusts the linear model so that the predicted distribution $P(Y \mid X_1, \ldots, X_n)$ closely matches the true distribution of Y given the inputs.

f) (1 point) Explain how many parameters would be needed to parameterize the model in (e) as a function of n and k.

Parameter Count for the Linear-Softmax Model

Recall that each variable X_i is one-hot encoded as a vector in $\mathbb{R}^k$. When concatenating n such vectors, the input vector becomes

$$\mathbf{x} \in \mathbb{R}^{nk}.$$

The linear model is defined as :

$$\mathbf{z} = W\mathbf{x} + \mathbf{b},$$

where

$$W \in \mathbb{R}^{k\times(nk)} \quad \text{and} \quad \mathbf{b} \in \mathbb{R}^k.$$

Number of Parameters

— The weight matrix W has k rows and nk columns, so it contains :

$$k \times (nk) = nk^2 \quad \text{parameters.}$$

— The bias vector $\mathbf{b}$ has k elements, adding :

$$k \quad \text{parameters.}$$

Thus, the total number of parameters is :

$$\boxed{nk^2 + k.}$$

Question 2 (English version) (5 points)

Give short answers for this section. Only a few lines of explaination are needed.

a) (1 points) What kind of activation functions were used for the intermediate layers of the original YOLO method and why ?

It uses LeakyReLUs. These activation functions provide a gradient of 1 in the active zone and a gradient of some constant value c when the input is less than zero. This can facilitate better gradient flow during backpropagation. b) (2 points) Why does a U-Net have skip connections and what is the motivation for the "U" structure of the architecture ?

Skip connections allow both gradient to flow more easily in the model, and allow information to be combined more easily across the network. The U structure allows coarse information to be recombined with fine scale information which is not possible in the usual "classic" CNN architectures (e.g. in the original LeNet and AlexNet style modeles.) c) (2 points) In your own words, why is the ADAM method so popular ? It is effective at replacing other heuristics for selecting learning rates. It has a reasonable number of hyperparameters which are relatively stable across models. The additional memory and computation required is negligible.

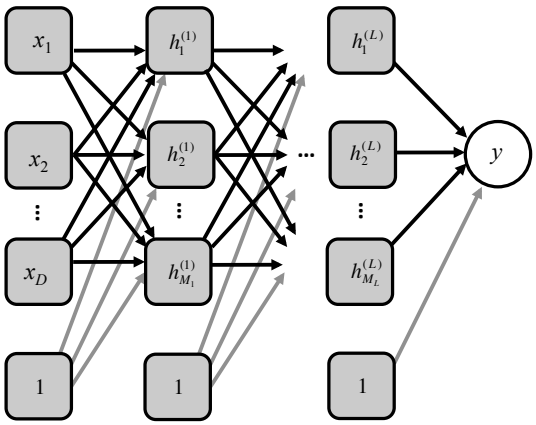

FIGURE 3 – Un réseau de neurones.

Question 3 (version française) (10 points)

Considérons un réseau de neurones avec une couche d'entrée comportant D unités, L couches cachées, chacune comportant $M_l = D$ unités, et une seule sortie binaire. Vous disposez de $i = 1, .., N$ exemples dans un ensemble d'entraînement où les étiquettes sont binaires $y_i \in \{0, 1\}$, et l'entrée est un vecteur de valeurs réelles continues $\mathbf{x}_i \in \mathbb{R}^D$. La fonction de perte est donnée par l'entropie croisée binaire :

$$L = -\sum_{i=1}^{N}\Big(y_i \log\big(f(\mathbf{x}_i)\big) + (1-y_i)\log\big(1-f(\mathbf{x}_i)\big)\Big), \quad f(\mathbf{x}_i) = f\Big(a^{(L+1)}\Big(\mathbf{h}^{(L)}\Big(\mathbf{a}^{(L)}\Big(\dots \mathbf{h}^{(1)}\Big(\mathbf{a}^{(1)}(\mathbf{x}_i)\Big)\Big)\Big)\Big)\Big). \tag{1}$$

où $\mathbf{x}_i$ est un exemple d'entrée, et y_i est une valeur scalaire continue correspondant à la cible de prédiction. La fonction d'activation de sortie f a la forme d'une sigmoïde et toutes les couches intermédiaires utilisent une sigmoïde **avec une connexion résiduelle**, de telle sorte que

$$f\Big(a^{(L+1)}(\mathbf{x}_i)\Big) = \frac{1}{1+\exp\Big(-a^{(L+1)}\Big(\mathbf{h}^{(L)}(\mathbf{x}_i)\Big)\Big)}, \qquad h_k^{(l)}\Big(a_k^{(l)}(\mathbf{x}_i)\Big) = \text{sigmoïde}\Big(a_k^{(l)}(\mathbf{x}_i)\Big) + a_k^{(l)}(\mathbf{x}_i), \tag{2}$$

où $\mathbf{a}^{(l)} = [a_1^{(l)} \dots\ a_K^{(l)}]^T$ est le vecteur résultant du calcul de la fonction de préactivation usuelle

$$\mathbf{a}^{(l)} = \mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)},$$

qui peut être simplifié en $\boldsymbol{\theta}^{(l)}\hat{\mathbf{h}}^{(l-1)}$ en utilisant l'astuce consistant à définir $\hat{\mathbf{h}}$ comme $\mathbf{h}$ avec un 1 concaténé à la fin.

a) (2 points) Qu'est-ce que $\frac{\partial \mathbf{h}^{(l)}}{\partial \mathbf{a}^{(l)}}$ dans ce modèle ?

On définit

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$

et

$$h_k^{(l)}\Big(a_k^{(l)}\Big) = \sigma\Big(a_k^{(l)}\Big) + a_k^{(l)}.$$

Alors, pour chaque unité k dans la couche l,

$$\frac{\partial h_k^{(l)}}{\partial a_k^{(l)}} = \sigma\Big(a_k^{(l)}\Big)\Big(1-\sigma\Big(a_k^{(l)}\Big)\Big) + 1.$$

Nous pouvons exprimer ceci sous forme matricielle. Soit la matrice diagonale

$$\mathbf{D} = \text{diag}\Big(\sigma(a_1^{(l)})(1-\sigma(a_1^{(l)})),\ \sigma(a_2^{(l)})(1-\sigma(a_2^{(l)})),\ \sigma(a_3^{(l)})(1-\sigma(a_3^{(l)}))\Big).$$

Alors, en utilisant la notation matricielle en majuscules, le jacobien s'écrit

$$\frac{\partial \mathbf{h}^{(l)}}{\partial \mathbf{a}^{(l)}} = \mathbf{I} + \mathbf{D},$$

où $\mathbf{I}$ est la matrice identité.

b) (2 points) Qu'est-ce que $\frac{\partial \mathbf{a}^{(l+1)}}{\partial \mathbf{h}^{(l)}}$? Utilisez la forme en « disposition dénominateur » pour le jacobien, définie par :

$$\left[\frac{\partial \mathbf{a}^{(l+1)}}{\partial \mathbf{h}^{(l)}}\right]_{ij} = \frac{\partial a_j^{(l+1)}}{\partial h_i^{(l)}}.$$

Selon la disposition dénominateur, nous avons

$$\frac{\partial \mathbf{a}^{(l+1)}}{\partial \mathbf{h}^{(l)}} = \mathbf{W}^{(l+1)\,T}.$$

c) (4 points) Comment pourrait-on écrire une expression compacte pour le gradient de la matrice de paramètres $\mathbf{W}^{(l)}$ à n'importe quelle couche en termes de l'« erreur » modifiée $\Delta^{(L+1)}$, calculée à la dernière couche ?

Posons

$$\Delta^{(l)} = \frac{\partial L}{\partial \mathbf{a}^{(l)}}$$

comme l'« erreur » à la couche l. Alors, la récurrence de la rétropropagation s'exprime par

$$\Delta^{(l)} = \left(\mathbf{I} + \mathbf{D}^{(l)}\right)\mathbf{W}^{(l+1)T}\Delta^{(l+1)},$$

où

$$\mathbf{D}^{(l)} = \operatorname{diag}\Big(\sigma(a_1^{(l)})(1-\sigma(a_1^{(l)})),\, \sigma(a_2^{(l)})(1-\sigma(a_2^{(l)})),\, \sigma(a_3^{(l)})(1-\sigma(a_3^{(l)}))\Big)$$

et $\sigma(a) = \frac{1}{1+\exp(-a)}$. Ensuite, le gradient de la matrice de paramètres à toute couche l s'exprime comme le produit extérieur de l'« erreur » à cette couche et de l'activation de la couche précédente :

$$\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \Delta^{(l)}\,\hat{\mathbf{h}}^{(l-1)T},$$

où $\hat{\mathbf{h}}^{(l-1)}$ est le vecteur d'activation augmenté de la couche $l-1$ (incluant un 1 constant si le biais est intégré via une représentation augmentée). De plus, en déroulant la récurrence depuis la couche de sortie $L+1$ jusqu'à n'importe quelle couche l, on obtient

$$\Delta^{(l)} = \left(\prod_{k=l}^{L}\Big[(\mathbf{I} + \mathbf{D}^{(k)})\,\mathbf{W}^{(k+1)T}\Big]\right)\Delta^{(L+1)}.$$

Ainsi, une expression compacte élégante pour le gradient à toute couche est

$$\boxed{\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \left(\prod_{k=l}^{L}\Big[(\mathbf{I} + \mathbf{D}^{(k)})\,\mathbf{W}^{(k+1)T}\Big]\right)\Delta^{(L+1)}\,\hat{\mathbf{h}}^{(l-1)T}.}$$

Cette formule exprime le gradient de la matrice de paramètres $\mathbf{W}^{(l)}$ en termes de l'« erreur » modifiée $\Delta^{(L+1)}$ calculée à la dernière couche, rétropropagée dans le réseau via les matrices $\mathbf{W}$ et les dérivées d'activation contenues dans $\mathbf{I} + \mathbf{D}$.

d) (2 points) En comparant et en contrastant cette expression avec la situation dans laquelle des sigmoïdes sans connexions résiduelles sont utilisées, comment les connexions résiduelles pourraient-elles améliorer la procédure d'optimisation ?

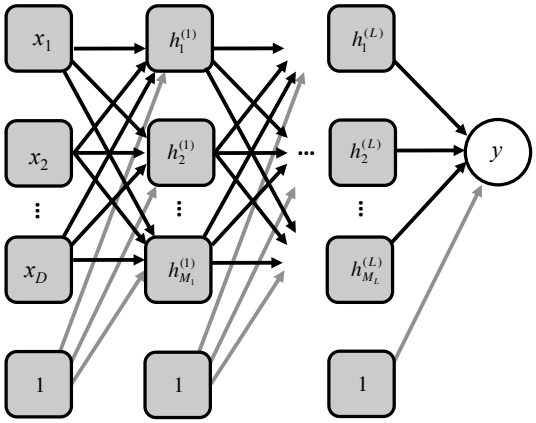

FIGURE 4 – A neural network.

Question 3 (English version) (10 points)

Consider a neural network with an input layer having D units, L hidden layers, each having $M_l = D$ units and a single binary output. You have $i = 1, .., N$ examples in a training set where the labels are binary $y_i \in \{0, 1\}$, the input is a vector of continuous real values $\mathbf{x}_i \in \mathbb{R}^D$. The loss function is given by the binary cross entropy :

$$L = -\sum_{i=1}^{N} \big(y_i \log(f(\mathbf{x}_i)) + (1 - y_i) \log(1 - f(\mathbf{x}_i)\big), f(\mathbf{x}_i) = f(a^{(L+1)}(\mathbf{h}^{(L)}(\mathbf{a}^{(L)}(\ldots \mathbf{h}^{(1)}(\mathbf{a}^{(1)}(\mathbf{x}_i)))))) \quad (3)$$

where $\mathbf{x}_i$ is an input example, y_i is a continuous scalar value which is the prediction target. The output activation function f has the form of a sigmoid and all the intermediate layers have a sigmoid **plus a residual connection**, such that

$$f(a^{(L+1)}(\mathbf{x}_i)) = \frac{1}{1 + \exp(-a^{(L+1)}(\mathbf{h}^{(L)}(\mathbf{x}_i)))}, \quad h_k^{(l)}(a_k{}^{(l)}(\mathbf{x}_i)) = \text{sigmoid}(a_k{}^{(l)}(\mathbf{x}_i)) + a_k{}^{(l)}(\mathbf{x}_i), \quad (4)$$

where $\mathbf{a}^{(l)} = [a_1^{(l)} ... \ a_K^{(l)}]^T$ is the vector resulting from the calculation of the usual preactivation function $\mathbf{a}^{(l)} = \mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}$, which could be simplified to $\boldsymbol{\theta}^{(l)}\hat{\mathbf{h}}^{(l-1)}$ using the trick of defining $\hat{\mathbf{h}}$ as $\mathbf{h}$ with a 1 concatenated at the end.

a) (2 points) What is $\frac{\partial \mathbf{h}^{(l)}}{\partial \mathbf{a}^{(l)}}$ in this model ?

Let

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

and

$$h_k^{(l)}(a_k^{(l)}) = \sigma(a_k^{(l)}) + a_k^{(l)}.$$

Then, for each unit k in layer l,

$$\frac{\partial h_k^{(l)}}{\partial a_k^{(l)}} = \sigma(a_k^{(l)})(1 - \sigma(a_k^{(l)})) + 1.$$

We can express this in matrix form. Let the diagonal matrix be

$$\mathbf{D} = \text{diag}\Big(\sigma(a_1^{(l)})(1 - \sigma(a_1^{(l)})),\ \sigma(a_2^{(l)})(1 - \sigma(a_2^{(l)})),\ \sigma(a_3^{(l)})(1 - \sigma(a_3^{(l)}))\Big).$$

Then, using bold uppercase notation for matrices, the Jacobian is given by

$$\frac{\partial \mathbf{h}^{(l)}}{\partial \mathbf{a}^{(l)}} = \mathbf{I} + \mathbf{D},$$

where $\mathbf{I}$ is the identity matrix.

b) (2 points) What is $\frac{\partial \mathbf{a}^{(l+1)}}{\partial \mathbf{h}^{(l)}}$? Use "denominator layout" form for the Jacobian, defined as :

$$\left[\frac{\partial \mathbf{a}^{(l+1)}}{\partial \mathbf{h}^{(l)}}\right]_{ij} = \frac{\partial a_j^{(l+1)}}{\partial h_i^{(l)}}.$$

Under denominator layout we have

$$\frac{\partial \mathbf{a}^{(l+1)}}{\partial \mathbf{h}^{(l)}} = \mathbf{W}^{(l+1)\,T}.$$

c) (4 points) How could one write a compact expression for the gradient of the parameter matrix $\mathbf{W}^{(l)}$ at any layer in terms of the modified "error" $\Delta^{(L+1)}$, computed at the last layer ?

Let

$$\Delta^{(l)} = \frac{\partial L}{\partial \mathbf{a}^{(l)}}$$

be the "error" at layer l. Then the backpropagation recurrence is given by

$$\Delta^{(l)} = \left(\mathbf{I} + \mathbf{D}^{(l)}\right)\mathbf{W}^{(l+1)T}\Delta^{(l+1)},$$

where

$$\mathbf{D}^{(l)} = \operatorname{diag}\Big(\sigma(a_1^{(l)})(1-\sigma(a_1^{(l)})),\, \sigma(a_2^{(l)})(1-\sigma(a_2^{(l)})),\, \sigma(a_3^{(l)})(1-\sigma(a_3^{(l)}))\Big)$$

and $\sigma(a) = \frac{1}{1+\exp(-a)}$. Then, the gradient of the parameter matrix at any layer l is expressed as the outer product of the "error" at that layer and the activation from the previous layer :

$$\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \Delta^{(l)}\,\hat{\mathbf{h}}^{(l-1)T},$$

where $\hat{\mathbf{h}}^{(l-1)}$ is the augmented activation vector from layer $l-1$ (including a constant 1 if you are incorporating the bias via an augmented representation). Moreover, unrolling the recurrence from the output layer $L+1$ to any layer l, we have

$$\Delta^{(l)} = \left(\prod_{k=l}^{L}\Big[\left(\mathbf{I} + \mathbf{D}^{(k)}\right)\mathbf{W}^{(k+1)T}\Big]\right)\Delta^{(L+1)}.$$

Thus, a "nice" compact expression for the gradient at any layer is

$$\boxed{\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \left(\prod_{k=l}^{L}\Big[\left(\mathbf{I} + \mathbf{D}^{(k)}\right)\mathbf{W}^{(k+1)T}\Big]\right)\Delta^{(L+1)}\,\hat{\mathbf{h}}^{(l-1)T}.}$$

This formula expresses the gradient of the parameter matrix $\mathbf{W}^{(l)}$ in terms of the modified "error" $\Delta^{(L+1)}$ computed at the final layer, propagated backward through the network via the matrices $\mathbf{W}$ and the activation derivatives contained in $\mathbf{I} + \mathbf{D}$.

d) (2 points) Comparing and contrasting this expression with the situation when sigmoids without residual connections are used, how might the residual connections improve the optimization procedure ?